공학박사 학위논문

Geometric Algorithms for Sensor Fusion: From Calibration to State Estimation

센서 퓨전을 위한 기하학적 알고리즘: 캘리브레이션과 상태추정

2018년 2월

서울대학교 대학원
기계항공공학부
강 동 훈

Geometric Algorithms for Sensor Fusion: From Calibration to State Estimation

센서 퓨전을 위한 기하학적 알고리즘: 캘리브레이션과 상태추정

지도교수 박 종 우

이 논문을 공학박사 학위논문으로 제출함

2017 년 12 월

서울대학교 대학원
기계항공공학부
강 동 훈

강동훈의 공학박사 학위논문을 인준함

2017 년 12 월

위 원 장 : 이 경 수

부위원장 : 박 종 우

위 원 : 이 동 준

위 원 : 오 성 회

위 원 : 김 진 욱

ABSTRACT

Geometric Algorithms for Sensor Fusion: Calibration and State Estimation

by

Donghoon Kang

School of Mechanical Engineering
Seoul National University

In this thesis, we present geometric algorithms for fusing measurements provided by various kinds of motion sensors such as cameras, inertial sensors, encoders, etc. Among many issues related to sensor fusion, we particularly consider the problem of two-frame sensor calibration, and estimation of attitudes and gyro bias.

Firstly, in the two-frame sensor calibration problem, the objective is to find

rigid-body homogeneous transformation matrices X, Y that best fit a set of equalities of the form $\mathbf{A}_i\mathbf{X} = \mathbf{Y}\mathbf{B}_i$, $i = 1, \ldots, N$, where $\{(\mathbf{A}_i, \mathbf{B}_i)\}$ are two sets of homogeneous transformation matrices obtained from two different sensor measurements. A fast and numerically robust local optimization algorithm for the two-frame sensor calibration objective function is proposed. Using coordinate-invariant differential geometric methods that take into account the matrix Lie group structure of the rigid-body transformations, our local descent method makes use of analytic gradients and Hessians, and a strictly descending fast step-size estimate to achieve significant performance improvements. Furthermore, we present a two-phase stochastic geometric optimization algorithm for finding a stochastic global minimizer based on our earlier local optimizer. Numerical simulation and real experiments demonstrate that our algorithm is superior to existing unit quaternion-based methods in terms of robustness and efficiency.

Secondly, we consider the problem of estimating attitudes and gyro bias by using inertial and magnetic sensors. To address this issue, we present an intrinsic unscented Kalman filtering (UKF) algorithm, of which novelty can be traced to the design of measurement function. In our formulation, the measurement has the form of $SO(3)$, which is given by the solution to Wahba's problem. Its merit is that the measurement noise covariance can consider the constraint on two direction vectors and is also well-defined with a full rank. Moreover, we present an offline algorithm for determining the parameters in this covariance from measurements of gravity and geomagnetic field by using actual accelerometers and magnetometers. Synthetic and real experiments show that our algorithm outperforms the existing state-of-the art estimators in terms of both convergence behavior and accuracy.

Keywords: geometric algorithm, sensor fusion, calibration, optimization, estimation

Student Number: 2012-30167

Contents

List of Tables

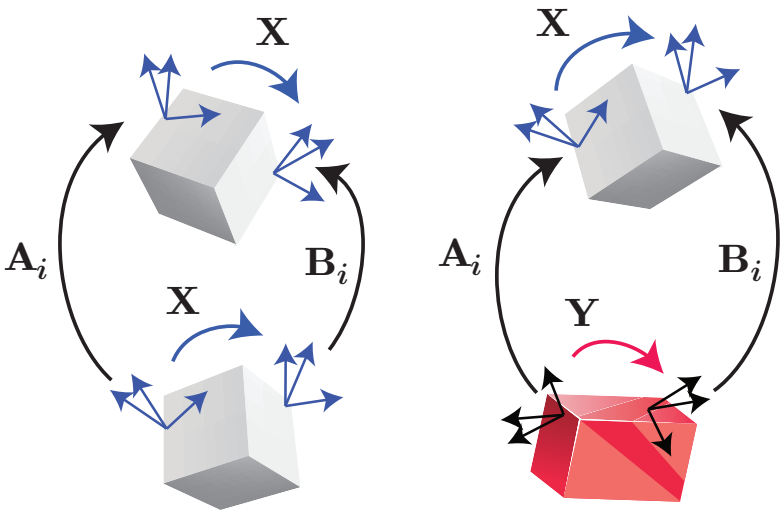
List of Figures

1 Introduction

1.1 Motivation of Research

Complex sensory systems such as robots or modern smartphones usually contain many different types of motion sensors including encoders, cameras, inertial and magnetic sensors. However, most of these sensors can provide only partial information about motions such as joint angles, images, angular velocities, accelerations and magnetic fields. To provide robust and complete information about motions of an object, these heterogeneous sensors need to be fused in a complementary manner.

Among many issues when fusing different types of motion sensors, we have focused on two fundamental problems: calibration and state estimation. More specifically, this thesis deals with three independent problems such as two-frame sensor calibration [1], the accuracy improvement of eye trackers [2], and the estimation of attitudes and gyro bias. In this thesis, we show that these problems can be resolved by using geometric algorithms in a consistent manner.



(a) One-frame problem (b) Two-frame problem

Figure 1.1: Sensor calibration problems involving (a) one frame and (b) two frames.

1.2 Literature Survey

1.2.1 Related Works: Two-Frame Sensor Calibration

The task of robots for sensing and acting usually requires an accurate knowledge of a robot geometry, particularly the relations between the frames of actuators and sensors. Figure 1.1 depicts the general context in which this problem arises. Given two rigid bodies, each with two reference frames attached at distinct locations, the relative displacements between each frame pair are respectively denoted $\mathbf{X}$ and $\mathbf{Y}$. Here $\mathbf{X}$ and $\mathbf{Y}$ are unknown; to determine them, various sensors are used to measure the displacements $\mathbf{A}_i, \mathbf{B}_i \in SE(3)$ of the frames at various

configurations of the two rigid bodies. Here $SE(3)$ denotes the Special Euclidean group of rigid body motions. A more detailed explanation about $SE(3)$ is given in the next section. If the two rigid bodies (and the locations of the attached reference frames) are identical, then $\mathbf{X} = \mathbf{Y}$, and the loop closure equation simplifies to $\mathbf{A}_i\mathbf{X} = \mathbf{X}\mathbf{B}_i$; this is the one-frame sensor calibration problem. In literature, the one-frame sensor calibration is often called as "hand-eye calibration".

The most well-known example of two-frame sensor calibration is the simultaneous hand-eye/robot-world calibration problem [3]. Hand-eye calibration alone is a one-frame calibration problem, as is the problem of robot-world calibration. There is extensive literature on solution methods for these two classical problems, which we describe in detail below. What is worth emphasizing is that with the recent proliferation of devices that integrate multiple sensors of different types, the two-frame calibration problem is being encountered in a variety of different contexts beyond the traditional industrial settings. For example, relative rotations between cameras and inertial measurement units (IMUs) on a mobile device along with their corresponding reference world frames must satisfy equations of the form $\mathbf{A}_i\mathbf{X} = \mathbf{Y}\mathbf{B}_i \in SO(3)$ as illustrated in Figure 3.1(a), where $SO(3)$ denotes the rotation group. Another application is aerial vehicle tracking (Figure 3.1(b)), in which an aerial vehicle equipped with an onboard camera identifies fixed markers in the environment while simultaneously being tracked by a ground-based camera.

There exists extensive literature on solution methods for the hand-eye and robot-world calibration problem. In [3] and [4], unit quaternion representations for rotations are used to develop linear least squares and iterative nonlinear methods for finding a local minimizer; in [5] dual quaternion representations for homogeneous transformations are used to find the rotation and translation components simultaneously, rather than in a decoupled way as in the previous two methods.

More closely related to our problem is the work on robotic hand-eye calibration in [3], [4] where the problem is formulated as follows:

$$\min_{\mathbf{q}_X, \mathbf{q}_Y \in \mathcal{S}^3} \sum_{i=1}^{N} \|\mathbf{q}_{A_i} \otimes \mathbf{q}_X - \mathbf{q}_Y \otimes \mathbf{q}_{B_i}\|^2, \tag{1.2.1}$$

where $(\mathbf{q}_{A_i}, \mathbf{q}_{B_i})$ are given unit quaternion representations of the measurements (A_i, B_i) and $(\mathbf{q}_X, \mathbf{q}_Y)$ denote the unknown unit quaternion representations of $(\mathbf{X}, \mathbf{Y})$. Here $\otimes$ denotes the multiplication operator for unit quaternion. A unit quaternion can be represented as a four-dimensional vector of unit norm (and can thus be regarded as a point on the three-sphere $\mathcal{S}^3$ in $\mathbb{R}^4$), and for every rotation R there exist two unit quaternion representations $\mathbf{q}$ and $-\mathbf{q}$. The objective function (1.2.1) can be reduced to a polynomial in degree two of the elements of $\mathbf{q}_X$ and $\mathbf{q}_Y$ (with the added constraint that $\mathbf{q}_X$ and $\mathbf{q}_Y$ are both of unit norm), and in principle closed form solutions can be obtained. However, because of the 2-1 property of unit quaternion, to find the global minimum of the objective function (1.2.1) one must try, in the event of N measurements $\{(\mathbf{q}_{A_i}, \mathbf{q}_{B_i})\}_{i=1,\ldots,N}$, up to 2^N possible solutions. The use of heuristics can reduce this number somewhat, but the additional bookkeeping and complexity of the formulas makes this method difficult to implement. Moreover, in [4] the closed-form solution for the rotation is ultimately used as an initial guess to a nonlinear optimization solver for the $SE(3)$ objective function, and the accuracy and performance of this numerical optimization procedure is even reported to be superior to the closed-form methods.

1.2.2 Related Works: Estimation of Attitude and Gyro Bias

A typical inertial measurement unit (IMU) consists of three-axis gyro, accelerometer, and magnetometer. Since the accelerometer and the magnetometer can respectively measure gravity and local geomagnetic field, the attitude of an IMU can be

determined without difficulty by solving Wahba's problem [6]. However, the solution to Wahba's problem given by only "two" direction measurements such as gravity and geomagnetic field can yield noisy attitude estimates. Furthermore, it is not responsive when the IMU is rotating. Angular rates are usually fused with two direction measurements. However, the angular rates measured by gyros are corrupted by slowly time-varying bias. Since the gyro bias severely affects the system stability, it is common to estimate the gyro bias as well as attitudes.

The problem of estimating attitudes and gyro bias is usually formulated by nonlinear equations. To resolve the problem, the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) is commonly used as a framework. Although the computational complexity of UKF is slightly higher than that of EKF, the UKF generally outperforms the EKF in terms of estimation accuracy [7, 8, 9]. Recently, the geometric, or intrinsic, versions of EKF [10, 11, 12, 13], UKF [14, 15], and particle filtering [16] algorithms evolving on matrix Lie groups have been presented. The merit of intrinsic filters is that their performance is independent of the choice of local coordinates, i.e., coordinate-invariant. However, when designing a particular intrinsic filer for estimating attitudes and gyro bias, one may find that there exist many choices on state space models. For example, Barrau's intrinsic EKF estimates the attitudes of an IMU, but it does not consider the gyro bias [12].

In research on attitude estimation with IMUs via various Kalman filters, there exist two types of measurement vectors: (i) Stacking two direction vectors $\mathbf{v}_1, \mathbf{v}_2$ (i.e., measurements of gravity and local geomagnetic field) into one column vector [17, 18] as follows

$$\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \mathbf{R}^T \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix},$$

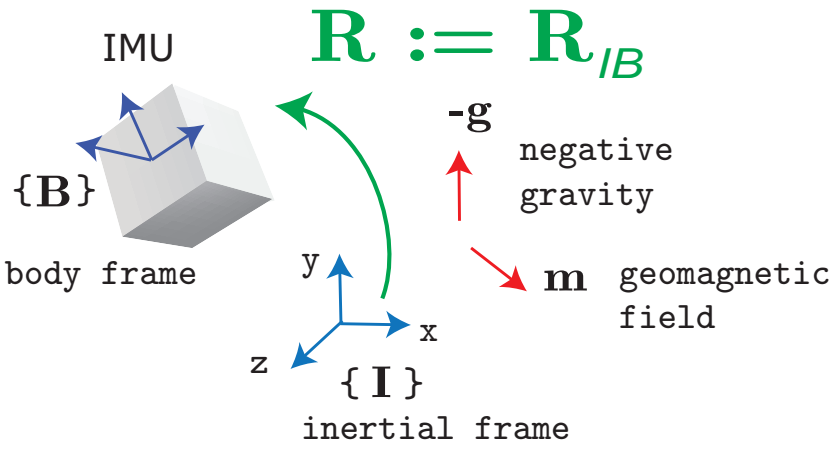


Figure 1.2: IMU body frame and inertial frame.

where $\mathbf{R}$ denotes the truth of an unknown rotation matrix and $\mathbf{w}_2, \mathbf{w}_2$ are zero-mean Gaussian noise vectors,

(ii) the reconstructed unit quaternion $\mathbf{q}_{measurement}$ given by the solution to Wahba's problem [19]

$$\mathbf{q}_{measurement} = \mathbf{q} + \boldsymbol{\eta}.$$

Here $\mathbf{q}$ is the truth of an unknown attitude quaternion and $\boldsymbol{\eta}$ denotes a zero-mean Gaussian noise vector.

However, these forms of measurement vectors have pitfalls as follows: the covariance of two direction vectors does not consider an underlying constraint i.e., the angle between gravity and local geomagnetic field is constant (see Figure 1.2). If a unit quaternion is assigned to the measurement vector, its covariance may suffer from the rank-deficiency owing to the additional unit norm constraint.

Compared to probabilistic approaches that usually employ various Kalman filters, deterministic nonlinear observers have been recently studied [6]. Among many promising nonlinear attitude observers, Mahony's nonlinear complementary filters

(NCFs) including the direct complementary filter, passive complementary filter, and explicit complementary filter are superior to other nonlinear attitude filters and almost ensure global stability of the observer error [20]. Although Mahony's NCFs adjusted by some constant scalar gains are fast and accurate, they do not fully consider the characteristics of IMU sensor noises. For this reason, NCFs cannot assure the optimality of the estimated attitude and gyro bias in terms of noise filtering.

1.3 Contributions of This Thesis

1.3.1 Two-Frame Sensor Calibration

• **Local optimization algorithm**

We first develop a fast and numerically robust local optimization algorithm that exploits the matrix Lie group structure of $SE(3)$. Our performance improvements are achieved in part by the availability of exact **analytic** gradients and Hessians of the objective function. We derive two specific local search methods, steepest descent and Newton's method. We also derive an analytic formula for rapidly computing a strictly descending **stepsize estimate**. Also as a by-product of our analysis, we characterize in a rigorous way the existence and uniqueness of solutions to the ideal case (i.e., no noise in the measurements).

• **Two-phase stochastic global optimization algorithm**

We also develop a two-phase stochastic global optimization algorithm, consisting of a global and local search phase, that extends the method of [21], [22] to the matrix Lie group $SE(3) \times SE(3)$ in a geometric way (i.e., invariant with respect

to choice of local coordinates, and also to left and right translations). The local search phase relies upon the local algorithm developed earlier. In the stochastic global search phase, we reduce the original unbounded search space $SE(3) \times SE(3)$ to the **compact space** $SO(3) \times SO(3)$, and generate random samples on this space in a coordinate-invariant way. An optimal Bayesian stopping criterion is used to terminate the algorithm, with probabilistic confidence levels obtained for the resulting solution to be a global optimizer.

• Application: affine registration for improving the accuracy of eye trackers

To improve the accuracy of 3-D eye position trackers using a **monocular** camera, we present a novel **compensation method** as a post-processing technique. We address the problem of determining an optimal registration function for fitting 3-D data consisting of the inaccurate estimates from the eye position tracker and their corresponding ground-truths. To obtain the ground-truths of 3-D eye positions, we propose two different systems by combining an optical motion capture system and checkerboards, which construct the form of two-frame sensor calibration. By solving a least-squares optimization problem, we can determine the optimal registration function in an affine form.

1.3.2 Unscented Kalman Filtering for Estimation of Attitude and Gyro Bias

• An accurate UKF algorithm for estimating attitudes and gyro bias

We present an accurate UKF algorithm for estimating attitudes and gyro bias in a coordinate-invariant way. The novelty of this algorithm can be traced to **the**

design of measurement function, which relates two different elements of matrix Lie groups, $SO(3) \times \mathbb{R}^3$ and $SO(3)$. Here $SO(3)$ denotes the rotation group and $\mathbb{R}^3$ represents a three-dimensional vector. In our formulation, the measurement has the form of $SO(3)$, which is given by the solution to Wahba's problem. Its merit is that the measurement noise covariance can consider the constraint on two direction vectors and is also well-defined with a full rank.

• An offline algorithm to determine the measurement noise covariance of the proposed UKF

An offline algorithm is proposed to determine the **measurement noise covariance** of the proposed UKF. By simply transforming the representation of the covariance from the IMU body-fixed frame into the ground-fixed frame, we find that the computation of the measurement noise covariance can be faster and more accurate when incorporated into the real-time attitude filter. Furthermore, we present an algorithm for determining the parameters in this covariance from measurements of gravity and geomagnetic field by using actual accelerometers and magnetometers.

1.4 Organization

This thesis is organized as follows. After reviewing geometric background in Chapter 2, we present geometric optimization algorithms to solve the two-frame sensor calibration problem in Chapter 3. As case studies, we examine (i) camera-marker calibration for unmanned aerial vehicle, (ii) head-eye calibration for humanoid robot, and (iii) affine registration for improving the accuracy of eye trackers. In Chapter 4, we present an intrinsic unscented Kalman filtering (UKF) algorithm as well as an offline algorithm for determining the parameters in the measurement noise covariance. In Chapter 5, we conclude this thesis by summarizing our main results.

2

Geometric Background

2.1 Matrix Lie Group

A matrix Lie group $\mathcal{G}$ is a group consisting of nonsingular matrices which is also a smooth manifold, such that multiplication and inversion are smooth. The Lie algebra $\mathfrak{g}$ associated with $\mathcal{G}$ is identified with the tangent space at the identity element in $\mathcal{G}$.

Given the matrix Lie group $\mathcal{G}$ and its associated Lie algebra $\mathfrak{g}$, the exponential mapping is the map $\exp : \mathfrak{g} \to \mathcal{G}$ defined by the matrix exponential: $\exp(\mathbf{y}) := \sum_{m=0}^{\infty} \frac{\mathbf{y}^m}{m!}$, where $\mathbf{y} \in \mathfrak{g}$. Over some neighborhood $\mathcal{N}$ of the identity in $\mathcal{G}$, the inverse of the exponential map, or logarithm $\log : \mathcal{N} \to \mathfrak{g}$, is defined as $\log(\mathbf{Y}) := \sum_{m=1}^{\infty} (-1)^{m+1} \frac{(\mathbf{Y}-\mathbf{I})^m}{m}$, where $\mathbf{Y} \in \mathcal{N}$ and $\mathbf{I}$ is the identity matrix. More rigorous discussions and details on Lie groups and their properties can be found in [23].

If the dimension of Lie algebra $\mathfrak{g}$ is n, i.e., $\dim \mathfrak{g} = n$, we can choose a set of basis matrices $\mathcal{E} = \{\mathbf{E}_1, \ldots, \mathbf{E}_n\}$ for $\mathfrak{g}$. Given $\mathcal{E}$, it is possible to identify an arbitrary element $\mathbf{y} \in \mathfrak{g}$ with a vector by defining linear "vee" operator $\vee : \mathfrak{g} \to \mathbb{R}^n$

via the identification $\mathbf{E}_i^\vee = \mathbf{e}_i \in \mathbb{R}^n$, where $\mathbf{e}_i$ is the i^{th} standard unit basis vector. For example, if $\mathbf{y} = \sum_{j=1}^n y_j \mathbf{E}_j$, then $\mathbf{y}^\vee := (y_1, \ldots, y_n)^T \in \mathbb{R}^n$. The inverse of $\vee$ is denoted by "wedge" operator $\wedge : \mathbb{R}^n \to \mathfrak{g}$, i.e. $(\mathbf{y}^\vee)^\wedge = \mathbf{y}$.

On matrix Lie algebra $\mathfrak{g}$, the Lie bracket $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is given by the matrix commutator: if $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathfrak{g}$, then $[\boldsymbol{\alpha}, \boldsymbol{\beta}] = \boldsymbol{\alpha}\boldsymbol{\beta} - \boldsymbol{\beta}\boldsymbol{\alpha}$. Given an element $\boldsymbol{\alpha} \in \mathfrak{g}$ and $\dim \mathfrak{g}^\vee = n$, the linear map $\mathrm{ad}_{\boldsymbol{\alpha}} : \mathfrak{g} \to \mathfrak{g}$ can be defined as $\mathrm{ad}_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) := [\boldsymbol{\alpha}, \boldsymbol{\beta}] \in \mathfrak{g}$, which also admits the matrix representation $[\mathrm{ad}_{\boldsymbol{\alpha}}] \in \mathbb{R}^{n \times n}$ as

$$[\mathrm{ad}_{\boldsymbol{\alpha}}]\, \boldsymbol{\beta}^\vee := ([\boldsymbol{\alpha}, \boldsymbol{\beta}])^\vee \in \mathbb{R}^n. \tag{2.1.1}$$

Given $\mathbf{X} \in \mathcal{G}$ and $\boldsymbol{\alpha} \in \mathfrak{g}$, the adjoint map $\mathrm{Ad}_{\mathbf{X}} : \mathfrak{g} \to \mathfrak{g}$ is defined as $\mathrm{Ad}_{\mathbf{X}}(\boldsymbol{\alpha}) := \mathbf{X}\boldsymbol{\alpha}\mathbf{X}^{-1}$. This also admits the matrix representation $[\mathrm{Ad}_{\mathbf{X}}] \in \mathbb{R}^{n \times n}$ given by

$$[\mathrm{Ad}_{\mathbf{X}}]\, \boldsymbol{\alpha}^\vee := (\mathbf{X}\boldsymbol{\alpha}\mathbf{X}^{-1})^\vee \in \mathbb{R}^n. \tag{2.1.2}$$

2.2 Geometry of Rigid Body Motions

Let us denote $SE(3)$ as the Special Euclidean group of rigid body motions of the form

$$\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.2.3}$$

where $\mathbf{R} \in SO(3)$ is a 3×3 rotation matrix (or an element of $SO(3)$, the special orthogonal group) and $\mathbf{p} \in \mathbb{R}^3$. Here $\mathbf{0}$ represents a zero vector, whose size is clear from the context.

$SO(3)$ is a matrix Lie group and its associated Lie algebra, denoted $so(3)$, is given by the set of 3×3 real skew-symmetric matrices with the matrix commutator as Lie bracket. We adopt the following notation: given $\mathbf{r} \in \mathbb{R}^3$, its 3×3 skew-symmetric representation is denoted $[\mathbf{r}] := \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$.

The exponential mapping from $so(3)$ to $SO(3)$ is given by the following formula: If $\mathbf{r} \neq \mathbf{0}$, then $\exp([\mathbf{r}]) \in SO(3)$ is given by

$$\exp([\mathbf{r}]) = \mathbf{I}_3 + \frac{\sin(\|\mathbf{r}\|)}{\|\mathbf{r}\|}[\mathbf{r}] + \frac{1-\cos(\|\mathbf{r}\|)}{\|\mathbf{r}\|^2}[\mathbf{r}]^2, \tag{2.2.4}$$

where $\|\cdot\|$ here denotes the standard Euclidean norm. In the trivial case $\mathbf{r} = \mathbf{0}$, we have $\exp([\mathbf{r}]) = \mathbf{I}_3$ where $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$ denotes the 3×3 identity matrix. The inverse of the exponential or logarithm, is also given as follows: Suppose $\mathbf{R} \in SO(3)$ such that $tr(\mathbf{R}) \neq -1$. Then

$$\log(\mathbf{R}) = \frac{\theta}{2\sin(\theta)}(\mathbf{R} - \mathbf{R}^T), \tag{2.2.5}$$

where θ satisfies $1 + 2\cos(\theta) = tr(\mathbf{R})$, $|\theta| < \pi$, and $\|\log(\mathbf{R})\| = \theta$. In the event that $tr(\mathbf{R}) = -1$, the logarithm $[\mathbf{r}] = \log(\mathbf{R})$ has two antipodal solutions $\pm\mathbf{r}$ which are determined from the relation $\mathbf{R} = \mathbf{I}_3 + (2/\pi^2)[\mathbf{r}]^2$.

We now provide useful formulas for the exponential and logarithm on $SE(3)$. Suppose $[\mathbf{r}] \in so(3)$ and $\mathbf{v} \in \mathbb{R}^3$. Then

$$\exp\left(\begin{bmatrix} [\mathbf{r}] & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{R} & \mathbf{G}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.2.6}$$

is an element of $SE(3)$, where $\mathbf{R} = \exp([\mathbf{r}]) \in SO(3)$ is given as in Eq.(2.2.4), and

$$\mathbf{G} = \mathbf{I}_3 + \frac{1-\cos(\|\mathbf{r}\|)}{\|\mathbf{r}\|^2}[\mathbf{r}] + \frac{\|\mathbf{r}\| - \sin(\|\mathbf{r}\|)}{\|\mathbf{r}\|^3}[\mathbf{r}]^2.$$

The logarithm on $SE(3)$ is given by the following formula: Let $\mathbf{R} \in SO(3)$ such that $tr(\mathbf{R}) \neq -1$, and let $\mathbf{p} \in \mathbb{R}^3$. Then

$$\log \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} [\mathbf{r}] & \mathbf{G}^{-1}\mathbf{p} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{2.2.7}$$

where $[\mathbf{r}] = \log(\mathbf{R})$ is given as in Eq.(2.2.5), and

$$\mathbf{G}^{-1} = \mathbf{I}_3 - \frac{1}{2}[\mathbf{r}] + \frac{2\sin(\|\mathbf{r}\|) - \|\mathbf{r}\|(1+\cos(\|\mathbf{r}\|))}{2\|\mathbf{r}\|^2\sin(\|\mathbf{r}\|)}[\mathbf{r}]^2.$$

Detailed explanations about these formulas are given in [24, 25].

2.3 Random Variables and Covariances

For any $\mathbf{T} \in \mathcal{G}$, the right translation map $\mathcal{R}_{\mathbf{T}} : \mathcal{G} \to \mathcal{G}$ and the left translation map $\mathcal{L}_{\mathbf{T}} : \mathcal{G} \to \mathcal{G}$ can be respectively defined as $\mathcal{R}_{\mathbf{T}}(\mathbf{X}) := \mathbf{X}\mathbf{T}$ and $\mathcal{L}_{\mathbf{T}}(\mathbf{X}) := \mathbf{T}\mathbf{X}$, where $\mathbf{X} \in \mathcal{G}$. Given $\mathbf{Y}, \mathbf{Z} \in \mathcal{G}$, let us define $g_r : \mathcal{G} \times \mathcal{G} \to \mathfrak{g}$ and $g_l : \mathcal{G} \times \mathcal{G} \to \mathfrak{g}$ as

$$g_r(\mathbf{Y}, \mathbf{Z}) = \log(\mathbf{Z}\mathbf{Y}^{-1})$$
$$g_l(\mathbf{Y}, \mathbf{Z}) = \log(\mathbf{Y}^{-1}\mathbf{Z}).$$

2.3.1 Right Invariant Covariance

Suppose $\boldsymbol{\eta} \in \mathbb{R}^n$ is a zero-mean Gaussian noise vector, i.e. $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\eta)$, where $\mathbf{P}_\eta$ denotes the covariance of $\boldsymbol{\eta}$. Then, we can define a random variable $\mathbf{X}$ for $\mathcal{G}$ according to

$$\mathbf{X} := \exp([\boldsymbol{\eta}])\, \mathbf{X}_0, \tag{2.3.8}$$

where $\mathbf{X}_0 \in \mathcal{G}$ is a constant matrix and $[\boldsymbol{\eta}] \in \mathfrak{g}$. We call $\boldsymbol{\eta}$ and $\mathbf{P}_\eta$ as the ***right invariant noise*** and the ***right invariant covariance*** of $\mathbf{X}$, respectively. The above terminologies stem from the invariance to the right translation map $\mathcal{R}_{\mathbf{T}}$ for all $\mathbf{T} \in \mathcal{G}$, which is given by

$$[\boldsymbol{\eta}] = g_r(\mathbf{X}_0, \mathbf{X}) = g_r(\mathcal{R}_{\mathbf{T}}(\mathbf{X}_0), \mathcal{R}_{\mathbf{T}}(\mathbf{X})).$$

Although the representation for random variables on matrix Lie groups given by Equation (2.3.8) is not rigorously defined, it has merit in terms of computational tractability [26].

2.3.2 Left Invariant Covariance

The random variable $\mathbf{X}$ in Equation (2.3.8) can be also represented as $\mathbf{X} = \mathbf{X}_0 \exp([\boldsymbol{\zeta}])$, where $[\boldsymbol{\zeta}] \in \mathfrak{m}$ and $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\zeta)$. Here $\mathbf{P}_\zeta$ denotes the covariance of the noise vector $\boldsymbol{\zeta}$. Note that $[\boldsymbol{\zeta}]$ is invariant to the left translation map $\mathcal{L}_\mathbf{T}$ for any $\mathbf{T} \in \mathcal{G}$ as follows: $[\boldsymbol{\zeta}] = g_l(\mathbf{X}_0, \mathbf{X}) = g_l(\mathcal{L}_\mathbf{T}(\mathbf{X}_0), \mathcal{L}_\mathbf{T}(\mathbf{X}))$, from which $\boldsymbol{\zeta}$ and $\mathbf{P}_\zeta$ are termed as the ***left invariant noise*** and the ***left invariant covariance*** of $\mathbf{X}$, respectively. From the relationship $\exp([\boldsymbol{\eta}])\,\mathbf{X}_0 = \mathbf{X}_0 \exp([\boldsymbol{\zeta}])$, we have

$$\boldsymbol{\eta} = [\mathrm{Ad}_{\mathbf{X}_0}]\,\boldsymbol{\zeta} \tag{2.3.9}$$

$$\mathbf{P}_\eta = [\mathrm{Ad}_{\mathbf{X}_0}]\,\mathbf{P}_\zeta\,[\mathrm{Ad}_{\mathbf{X}_0}]^T. \tag{2.3.10}$$

Our algorithm has been developed by using *right invariant noises* and *right invariant covariances*, while left invariant noises and left invariant covariances may also be employed.

3

Two-Frame Sensor Calibration

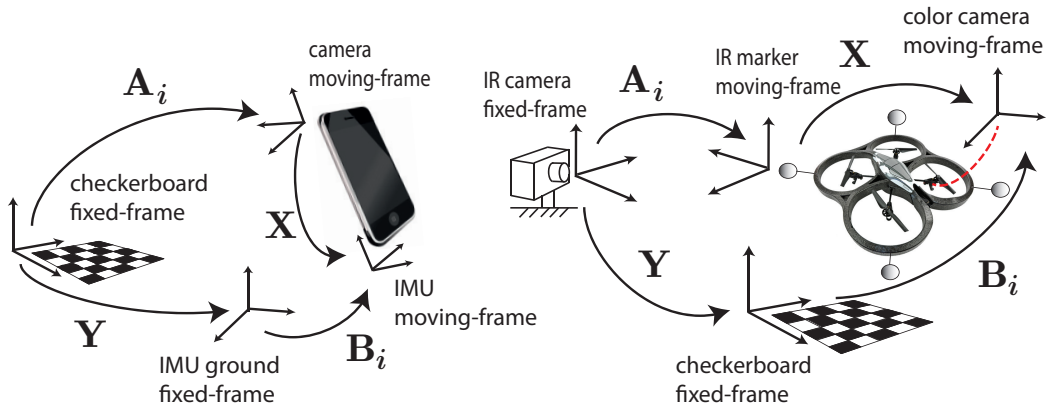
3.1 Introduction

In this chapter, we address the following version of the **"two-frame sensor calibration"** problem: given pairs of homogeneous rigid-body transformation matrices $\{(\mathbf{A}_i, \mathbf{B}_i)\}$, $i = 1, \ldots, N$, where each $\mathbf{A}_i$ and $\mathbf{B}_i$ is a 4×4 homogeneous rigid-body transformation matrix belonging to the Special Euclidean group $SE(3)$ [27] of rigid-body motions, find $\mathbf{X}, \mathbf{Y} \in SE(3)$ that is a best fit to the N matrix equalities $\mathbf{A}_i\mathbf{X} = \mathbf{Y}\mathbf{B}_i$, $i = 1, \ldots, N$. In its most common form, this problem is formulated as a least-squares optimization problem over $\mathbf{X}, \mathbf{Y} \in SE(3)$, in which the objective function is given by

$$\min_{\mathbf{X},\mathbf{Y} \in SE(3)} \sum_i \|\mathbf{A}_i\mathbf{X} - \mathbf{Y}\mathbf{B}_i\|^2, \tag{3.1.1}$$

where $\|\cdot\|$ denotes the Frobenius norm, i.e., $\|\mathbf{A}\| = \sqrt{\mathrm{tr}(\mathbf{A}\mathbf{A}^T)}$ for a matrix $\mathbf{A}$. Here $tr(\cdot)$ denotes the trace of a matrix.

In the absence of measurement noise, the loop closure equation $\mathbf{A}_i\mathbf{X} = \mathbf{Y}\mathbf{B}_i$



(a) smartphone with a camera and an IMU (b) UAV with a camera and IR markers

Figure 3.1: Examples of the two-frame calibration problem.

must be satisfied for all measurements $i = 1, \ldots, N$.

The objective function (3.1.1) is in general nonconvex, and possesses local minima whose number and properties are strongly influenced by the level of noise in the measurements $\{(\mathbf{A}_i, \mathbf{B}_i)\}$.

3.2 Existence and Uniqueness of Solutions to $\mathbf{AX} = \mathbf{YB}$

The equation $\mathbf{AX} = \mathbf{YB}$ on $SE(3)$ can be expressed as the pair of equations

$$\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_B$$
$$\mathbf{R}_A\mathbf{p}_X + \mathbf{p}_A = \mathbf{R}_Y\mathbf{p}_B + \mathbf{p}_Y,$$

where $\mathbf{R}_{(\cdot)} \in SO(3)$ and $\mathbf{p}_{(\cdot)} \in \mathbb{R}^3$ represent the rotational part and translational part of $(\cdot)$ on $SE(3)$. Here, $\mathbf{R}_A, \mathbf{R}_B \in SO(3)$ and $\mathbf{p}_A, \mathbf{p}_B \in \mathbb{R}^3$ are given while $\mathbf{R}_X, \mathbf{R}_Y \in SO(3)$ and $\mathbf{p}_X, \mathbf{p}_Y \in \mathbb{R}^3$ are unknown. In [3], it is shown that any

solution to the above equations must satisfy a corresponding set of linear equations, but the existence and uniqueness of solutions is not precisely characterized. In this section, we make mathematically precise statements about the existence and uniqueness of solutions $(\mathbf{X}, \mathbf{Y})$ to the problem given by $\mathbf{AX} = \mathbf{YB}$; detailed proofs of the ensuing propositions are provided in Appendix A.1.

3.2.0.1 Existence and Uniqueness of Solutions to $\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_B$ on $SO(3)$

Given $\mathbf{R}_A, \mathbf{R}_B \in SO(3)$, we now consider the existence and uniqueness of solutions to the equation $\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_B$, where $\mathbf{R}_X, \mathbf{R}_Y \in SO(3)$ are unknown.

Proposition 3.1. *Given two rotation matrix pairs* $\{(\mathbf{R}_{A_1}, \mathbf{R}_{B_1}), (\mathbf{R}_{A_2}, \mathbf{R}_{B_2})\}$*, we consider a pair of equations*

$$\mathbf{R}_{A_1}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_1}, \quad \mathbf{R}_{A_2}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_2}, \tag{3.2.2}$$

where $\mathbf{R}_X, \mathbf{R}_Y \in SO(3)$ *are unknown. Let us define* $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^3$ *as follows:*

$$\begin{aligned} [\boldsymbol{\alpha}] &= \log(\mathbf{R}_{A_1}\mathbf{R}_{A_2}^T) \\ [\boldsymbol{\beta}] &= \log(\mathbf{R}_{B_2}^T\mathbf{R}_{B_1}). \end{aligned}$$

If $\|\boldsymbol{\alpha}\| = \|\boldsymbol{\beta}\|$*, then Equations (3.2.2) have a one-parameter family of solutions* $(\mathbf{R}_X, \mathbf{R}_Y)_t$ *given by*

$$\mathbf{R}_X = \mathbf{R}_{A_2}^T \exp([\boldsymbol{\alpha}]t)\boldsymbol{\Theta}_p = \mathbf{R}_{A_2}^T\boldsymbol{\Theta}_p \exp([\boldsymbol{\beta}]t) \tag{3.2.3}$$

$$\mathbf{R}_Y = \exp([\boldsymbol{\alpha}]t)\boldsymbol{\Theta}_p\mathbf{R}_{B_2}^T = \boldsymbol{\Theta}_p \exp([\boldsymbol{\beta}]t)\mathbf{R}_{B_2}^T, \tag{3.2.4}$$

where $t \in [0, 2\pi]$*, and* $\boldsymbol{\Theta}_p \in SO(3)$ *is any particular solution to* $\Theta\boldsymbol{\beta} = \boldsymbol{\alpha}$*.*

Proposition 3.2. *Given three rotation pairs* $(\mathbf{R}_{A_i}, \mathbf{R}_{B_i})$*,* $i = 1, 2, 3$*, we can define*

$$\begin{aligned} [\boldsymbol{\alpha}_{jk}] &= \log(\mathbf{R}_{A_j}\mathbf{R}_{A_k}^T) \\ [\boldsymbol{\beta}_{jk}] &= \log(\mathbf{R}_{B_k}^T\mathbf{R}_{B_j}), \end{aligned}$$

and 3×3 *matrices*

$$\begin{aligned} \mathbf{\Phi} &= \begin{bmatrix} \boldsymbol{\alpha}_{21} & \boldsymbol{\alpha}_{31} & \boldsymbol{\alpha}_{21} \times \boldsymbol{\alpha}_{31} \end{bmatrix} \\ \mathbf{\Psi} &= \begin{bmatrix} \boldsymbol{\beta}_{21} & \boldsymbol{\beta}_{31} & \boldsymbol{\beta}_{21} \times \boldsymbol{\beta}_{31} \end{bmatrix}. \end{aligned}$$

Then there exists a unique solution pair $(\mathbf{R}_X, \mathbf{R}_Y)$ *to the set of equations*

$$\mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_i} \ , \ i = 1, 2, 3,$$

given by

$$\mathbf{R}_X = \mathbf{R}_{A_1}^T \Phi \Psi^{-1}, \quad \mathbf{R}_Y = \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{R}_{B_1}, \tag{3.2.5}$$

if and only if both matrices Φ *and* Ψ *are nonsingular and* $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{\Psi}^T\mathbf{\Psi}$.

3.2.0.2 Existence and Uniqueness of Solutions $(\mathbf{X}, \mathbf{Y})$ to $\mathbf{AX} = \mathbf{YB}$ on $SE(3)$

Proposition 3.3. *Given two rigid-body transformation matrix pairs* $\{(\mathbf{A}_1, \mathbf{B}_1), (\mathbf{A}_2, \mathbf{B}_2)\}$, *consider the pair of equations*

$$\mathbf{A}_1\mathbf{X} = \mathbf{Y}\mathbf{B}_1, \quad \mathbf{A}_2\mathbf{X} = \mathbf{Y}\mathbf{B}_2, \tag{3.2.6}$$

where $\mathbf{X}, \mathbf{Y} \in SE(3)$ *are unknown. Define* $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^3$ *and* $\mathbb{A} \in \mathbb{R}^{6\times 6}$ *as follows:*

$$\begin{aligned} [\boldsymbol{\alpha}] &= \log(\mathbf{R}_{A_1}\mathbf{R}_{A_2}^T) \\ [\boldsymbol{\beta}] &= \log(\mathbf{R}_{B_2}^T\mathbf{R}_{B_1}). \\ \mathbb{A} &= \begin{bmatrix} \mathbf{R}_{A_1} & -\mathbf{I} \\ \mathbf{R}_{A_2} & -\mathbf{I} \end{bmatrix} \end{aligned}$$

If $\|\boldsymbol{\alpha}\| = \|\boldsymbol{\beta}\|$, *then Equations (3.2.6) have a* $(7 - rank(\mathbb{A}))$*-parameter family of solutions* $(\mathbf{X}, \mathbf{Y})$.

Proposition 3.4. *Given the three rigid-body motion pairs* $(\mathbf{A}_i, \mathbf{B}_i)$, $i = 1, 2, 3$, *and corresponding rotation pairs* $(\mathbf{R}_{A_i}, \mathbf{R}_{B_i})$ *and translation pairs* $(\mathbf{p}_{A_i}, \mathbf{p}_{B_i})$, *define the same* $\mathbf{\Phi}$ *and* $\mathbf{\Psi}$ *as in Proposition 3.2 and*

$$\begin{aligned} \mathbb{A} &= \begin{bmatrix} \mathbf{R}_{A_1} & -\mathbf{I} \\ \mathbf{R}_{A_2} & -\mathbf{I} \\ \mathbf{R}_{A_3} & -\mathbf{I} \end{bmatrix} \in \mathbb{R}^{9\times 6} \\ \boldsymbol{\eta} &= \begin{bmatrix} \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{R}_{B_1}\mathbf{p}_{B_1} - \mathbf{p}_{A_1} \\ \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{R}_{B_2}\mathbf{p}_{B_2} - \mathbf{p}_{A_2} \\ \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{R}_{B_3}\mathbf{p}_{B_3} - \mathbf{p}_{A_3} \end{bmatrix} \in \mathbb{R}^9. \end{aligned}$$

Then there exists a unique solution of $(\mathbf{X}, \mathbf{Y})$ *if and only if both* $\mathbf{\Phi}$ *and* $\mathbf{\Psi}$ *are nonsingular with* $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{\Psi}^T\mathbf{\Psi}$, $rank(\mathbb{A}) = 6$, *and* $\boldsymbol{\eta}$ *is linearly dependent on the column vectors of* $\mathbb{A}$.

Propositions 3.1 and 3.2 describe the existence and uniqueness of solutions when noise-free measurements on the rotation group $SO(3)$ are available. Propositions 3.3 and 3.4 are more general versions of the existence and uniqueness conditions for the solution on the Euclidean group $SE(3)$ under the assumption of noise-free measurements.

3.3 Local Least Squares Minimization

3.3.1 Least Squares Objective Function

In practice there will not exist an exact solution to $\mathbf{A}_i\mathbf{X} = \mathbf{Y}\mathbf{B}_i$ since the measurements $\mathbf{A}_i$ and $\mathbf{B}_i$ are corrupted by sensor noise. In this section, we consider

the minimization of the following least squares criterion:

$$\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\mathbf{X}-\mathbf{Y}\mathbf{B}_i\|^2$$

where several choices of $\|\,\mathbf{.}\,\|$ are available. Here, we define $\|\,\mathbf{.}\,\|^2$ as $\|\mathbf{P}-\mathbf{Q}\|^2 = \|\mathbf{R}_P-\mathbf{R}_Q\|_F^2+\zeta\|\mathbf{p}_P-\mathbf{p}_Q\|^2$ where $\mathbf{R}_P,\mathbf{R}_Q\in SO(3)$ and $\mathbf{p}_P,\mathbf{p}_Q\in\mathbb{R}^3$ are the rotations and the translations of $\mathbf{P},\mathbf{Q}\in SE(3)$ respectively. Here, $\|\,\mathbf{.}\,\|_F$ denotes the Frobenius norm and $\zeta\in\mathbb{R}_+$ is a weighting factor for the translation error. The least squares criterion becomes as follows:

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\mathbf{X}-\mathbf{Y}\mathbf{B}_i\|^2 = \frac{1}{2}\sum_{i=1}^{N}(6-2tr(\mathbf{R}_X^T\mathbf{R}_{A_i}^T\mathbf{R}_Y\mathbf{R}_{B_i}) \\ +\zeta\|\mathbf{R}_{A_i}\mathbf{p}_X+\mathbf{p}_{A_i}-\mathbf{R}_Y\mathbf{p}_{B_i}-\mathbf{p}_Y\|^2).\end{aligned} \tag{3.3.7}$$

The above Equation (3.3.7) can be established using the general matrix trace identity $tr(\mathbf{ABC}) = tr(\mathbf{CAB}) = tr(\mathbf{BCA})$ for matrices $\mathbf{A},\mathbf{B},\mathbf{C}$. As stated in the previous section, this problem is not a convex nor even a quasi-convex problem, typically possessing multiple local minima. Applying stochastic global optimization techniques to this problem directly is problematic because of the infinite volume of the search space $SE(3)\times SE(3)$ resulting from the unbounded space of pure translations in $SE(3)$.

We now show that the function can be easily reduced to a quadratic function on $SO(3)\times SO(3)$, which from the compactness of $SO(3)$ has a search space of bounded volume. To see why, first note that

$$\min_{\mathbf{X},\mathbf{Y}\in SE(3)}\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\mathbf{X}-\mathbf{Y}\mathbf{B}_i\|^2 = \min_{\mathbf{R}_X,\mathbf{R}_Y\in SO(3)}\left(\min_{\mathbf{p}_X,\mathbf{p}_Y\in\mathbb{R}^3}\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\mathbf{X}-\mathbf{Y}\mathbf{B}_i\|^2\right).$$

Since $\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{A}_i\mathbf{X}-\mathbf{Y}\mathbf{B}_i\|^2$ is a least squares criterion, it is a convex quadratic

function with respect to $\mathbf{p}_X$ and $\mathbf{p}_Y$ in which closed-form solutions to the sub-minimization $\min_{\mathbf{p}_X,\mathbf{p}_Y\in\mathbb{R}^3} \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{X} - \mathbf{Y}\mathbf{B}_i\|^2$ are available. Note that the closed-form solutions $\mathbf{p}_X^*(\mathbf{R}_X, \mathbf{R}_Y)$ and $\mathbf{p}_Y^*(\mathbf{R}_X, \mathbf{R}_Y)$ are functions of $\mathbf{R}_X$ and $\mathbf{R}_Y$. By substituting these into Equation (3.3.7), we can define

$$\begin{aligned} J(\mathbf{R}_X, \mathbf{R}_Y) &= \min_{\mathbf{p}_X,\mathbf{p}_Y\in\mathbb{R}^3} \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{X} - \mathbf{Y}\mathbf{B}_i\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{N}(6 - 2tr(\mathbf{R}_X^T\mathbf{R}_{A_i}^T\mathbf{R}_Y\mathbf{R}_{B_i}) + \zeta\|\mathbf{R}_{A_i}\mathbf{p}_X^* + \mathbf{p}_{A_i} - \mathbf{R}_Y\mathbf{p}_{B_i} - \mathbf{p}_Y^*\|^2). \end{aligned}$$

The entire problem now reduces to

$$\min_{\mathbf{R}_X,\mathbf{R}_Y\in SO(3)} J(\mathbf{R}_X, \mathbf{R}_Y), \tag{3.3.8}$$

which is a minimization on $SO(3)\times SO(3)$. We finally derive the following expression for $J(\mathbf{R}_X, \mathbf{R}_Y)$:

$$J(\mathbf{R}_X, \mathbf{R}_Y) = \frac{1}{2}\sum_{i=1}^{18} \lambda_i \left(tr(\mathbf{P}_i\mathbf{R}_X) + tr(\mathbf{Q}_i\mathbf{R}_Y)\right)^2 + tr(\mathbf{P}_0\mathbf{R}_X) + tr(\mathbf{Q}_0\mathbf{R}_Y) + c, \tag{3.3.9}$$

where $\mathbf{P}_i, \mathbf{Q}_i \in \mathbb{R}^{3\times 3}$ and $\lambda_i, c \in \mathbb{R}$ are obtained by the eigenvalue analysis of the original function (see Appendix A.2). Note that the minimization (3.3.8) is not a cyclic minimization (namely, it does not optimize over the variables in a cyclical fashion, by optimizing over one variable while keeping the remaining fixed, and iterating this procedure over all variables). We remark that the time complexity of the function evaluation of $J(\mathbf{R}_X, \mathbf{R}_Y)$ reduces from $O(N)$ to $O(1)$. This leads to gradient and Hessian evaluation algorithms with complexity $O(1)$ as well, greatly enhancing our stochastic global optimization algorithm by reducing the function evaluation times for a large number of sample points.

3.3.2 Determining the Initial Guess $(\mathbf{R}_{X_0}, \mathbf{R}_{Y_0})$

From the rotational parts $\{(\mathbf{R}_{A_i}, \mathbf{R}_{B_i})\}_{i=1,\ldots,N}$ of pose data pairs $\{(\mathbf{A}_i, \mathbf{B}_i)\}_{i=1,\ldots,N}$, one can choose two independent equations with any $k \in [1, N]$, e.g., $\mathbf{R}_{A_k}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_k}$ and $\mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_i}$ $(i \neq k)$. From the above two equations, $\mathbf{R}_Y$ can be eliminated by $\mathbf{R}_{A_k}^T\mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_X\mathbf{R}_{B_k}^T\mathbf{R}_{B_i}$. Let us denote $[\boldsymbol{\alpha}_{1i}] \triangleq \log(\mathbf{R}_{A_k}^T\mathbf{R}_{A_i})$ and $[\boldsymbol{\beta}_{1i}] \triangleq \log(\mathbf{R}_{B_1}^T\mathbf{R}_{B_i})$, so that the above reduces to $\mathbf{R}_X\boldsymbol{\beta}_{1i} = \boldsymbol{\alpha}_{1i}$. In the literature [24], a closed-form solution for the minimum of $\sum \|\mathbf{R}_{A_i}\mathbf{R}_X - \mathbf{R}_X\mathbf{R}_{B_i}\|^2$ is given, in which

$$\mathbf{R}_X = (\mathbf{U}^T\mathbf{U})^{-1/2}\mathbf{U}^T, \tag{3.3.10}$$

where $\mathbf{U} = \sum_i \boldsymbol{\beta}_{1i}\boldsymbol{\alpha}_{1i}^T$. In a similar fashion, one can eliminate $\mathbf{R}_X$ rather than $\mathbf{R}_Y$ to obtain $\mathbf{R}_{A_k}\mathbf{R}_{A_i}^T\mathbf{R}_Y = \mathbf{R}_Y\mathbf{R}_{B_k}\mathbf{R}_{B_i}^T$. Let us define $\tilde{\boldsymbol{\beta}}_{1i}$ and $\tilde{\boldsymbol{\alpha}}_{1i}$ as $[\tilde{\boldsymbol{\alpha}}_{1i}] \triangleq \log(\mathbf{R}_{A_k}\mathbf{R}_{A_i}^T)$ and $[\tilde{\boldsymbol{\beta}}_{1i}] \triangleq \log(\mathbf{R}_{B_k}\mathbf{R}_{B_i}^T)$. Applying the same closed-form solution, $\mathbf{R}_Y$ can be obtained by

$$\mathbf{R}_Y = (\mathbf{V}^T\mathbf{V})^{-1/2}\mathbf{V}^T, \tag{3.3.11}$$

where $\mathbf{V} = \sum_i \tilde{\boldsymbol{\beta}}_{1i}\tilde{\boldsymbol{\alpha}}_{1i}^T$. In this way one can obtain a reasonable set of initial values $(\mathbf{R}_{X_0}, \mathbf{R}_{Y_0})$ for $(\mathbf{R}_X, \mathbf{R}_Y)$.

3.3.3 Local Geometric Minimization

The objective function (3.3.9) is minimized through generalizations of the steepest descent algorithm and Newton's method to the search space $SO(3) \times SO(3)$. It is instructive to first review the standard vector space versions of these algorithms. Given a twice-differentiable objective function $J(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, the steepest descent algorithm applies the following iteration until a suitable convergence criterion is met:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + m_k\mathbf{d}_k,$$

where the search direction $\mathbf{d}_k \in \mathbb{R}^n$ is taken to be the gradient of $J(\mathbf{x})$ at $\mathbf{x}_k$, i.e.,

$$\mathbf{d} = -\nabla J(\mathbf{x}_k),$$

and the stepsize m_k is a positive scalar, typically chosen so as to minimize J along the search direction:

$$m_k = \arg\min_{m \in \mathbb{R}} J(\mathbf{x} + m_k \mathbf{d}_k). \tag{3.3.12}$$

In Newton's method, the search direction is taken to be

$$\mathbf{d} = -[\nabla^2 J(\mathbf{x}_k)]^{-1} \nabla J(\mathbf{x}_k),$$

where $\nabla^2 J(\mathbf{x}_k) \in \mathbb{R}^{n \times n}$ denotes the Hessian of J at $\mathbf{x}_k$ (recall that the gradient and Hessian correspond to the first- and second-order terms in the Taylor series expansion of $J(\mathbf{x})$:

$$J(\mathbf{x} + \mathbf{h}) = J(\mathbf{x}) + \nabla J(\mathbf{x})\mathbf{h} + \frac{1}{2}\mathbf{h}^T \nabla^2 J(\mathbf{x})\mathbf{h} + \dots \ .$$

The following sections present a geometric generalization of these standard vector space optimization methods. In a geometric space like $SO(3) \times SO(3)$, the straight lines in the stepsize computation (3.3.12) are now replaced by minimal geodesics. Not only exact analytic gradients and Hessians that enhance the performance and convergence of local optimization algorithms are derived in Section 3.3.3.1, but also "strictly descending stepsize estimate" that enables one to use a traditional stochastic optimization algorithm in our geometric search space is presented in Section 3.3.3.2. The matrix Lie group structure of $SO(3)$ is then exploited to generate a geodesic curve along the direction and update the state on the curve with strictly descending stepsize estimate.

3.3.3.1 Gradient and Hessian Formulas

Because the objective function (3.3.9) is defined on $SO(3) \times SO(3)$, which as is well-known is not a vector space, appropriate notions of the gradient and Hessian are needed. We first expand $(\mathbf{R}_X, \mathbf{R}_Y)$ about $(\mathbf{R}_{X_k}, \mathbf{R}_{Y_k})$ via the matrix exponential as follows:

$$\mathbf{R}_X = \mathbf{R}_{X_k}(\mathbf{I} + [\boldsymbol{\omega}_{\mathbf{R}_X}] + \frac{1}{2}[\boldsymbol{\omega}_{\mathbf{R}_X}]^2 + \ldots)$$
$$\mathbf{R}_Y = \mathbf{R}_{Y_k}(\mathbf{I} + [\boldsymbol{\omega}_{\mathbf{R}_Y}] + \frac{1}{2}[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2 + \ldots).$$

$J(\mathbf{R}_X, \mathbf{R}_Y)$ can now expanded up to second order and the gradient and Hessian can be obtained by differentiating it with respect to $\boldsymbol{\omega}_{\mathbf{R}_X}$ and $\boldsymbol{\omega}_{\mathbf{R}_Y}$; the constant and first-order terms then correspond to the gradient and Hessian, respectively. The specific derivations and formulas are given in Appendix A.3.

3.3.3.2 Strictly Descending Stepsize

We first expand $(\mathbf{R}_X, \mathbf{R}_Y)$ about $(\mathbf{R}_{X_k}, \mathbf{R}_{Y_k})$ via the exponential mapping as follows:

$$\mathbf{R}_X = \mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} = \mathbf{R}_{X_k}(\mathbf{I} + \frac{\sin\|\boldsymbol{\omega}_{\mathbf{R}_X}\|t}{\|\boldsymbol{\omega}_{\mathbf{R}_X}\|}[\boldsymbol{\omega}_{\mathbf{R}_X}] + \frac{1-\cos\|\boldsymbol{\omega}_{\mathbf{R}_X}\|t}{\|\boldsymbol{\omega}_{\mathbf{R}_X}\|^2}[\boldsymbol{\omega}_{\mathbf{R}_X}]^2)$$
$$\mathbf{R}_Y = \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t} = \mathbf{R}_{Y_k}(\mathbf{I} + \frac{\sin\|\boldsymbol{\omega}_{\mathbf{R}_Y}\|t}{\|\boldsymbol{\omega}_{\mathbf{R}_Y}\|}[\boldsymbol{\omega}_{\mathbf{R}_Y}] + \frac{1-\cos\|\boldsymbol{\omega}_{\mathbf{R}_Y}\|t}{\|\boldsymbol{\omega}_{\mathbf{R}_Y}\|^2}[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2)$$

where $t \in \mathbb{R}$ is the stepsize variable and $(\boldsymbol{\omega}_{\mathbf{X}}, \boldsymbol{\omega}_{\mathbf{Y}})$ is the given search direction. The line search procedure is then given by

$$t^* = \arg\min_{t \in \mathbb{R}} \phi(t), \tag{3.3.13}$$

where $\phi(t) = J(\mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t}, \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}) = c_0 + c_1 \sin t_X + c_2 \cos t_X + c_3 \sin t_Y + c_4 \cos t_Y + c_5 \sin t_X \sin t_Y + c_6 \sin t_X \cos t_Y + c_7 \cos t_X \sin t_Y + c_8 \cos t_X \cos t_Y$ with

$\hat{\boldsymbol{\omega}}_{\mathbf{R}_X} = \boldsymbol{\omega}_{\mathbf{R}_X}/\|\boldsymbol{\omega}_{\mathbf{R}_X}\|$, $\hat{\boldsymbol{\omega}}_{\mathbf{R}_Y} = \boldsymbol{\omega}_{\mathbf{R}_Y}/\|\boldsymbol{\omega}_{\mathbf{R}_Y}\|$, $t_X = \|\boldsymbol{\omega}_{\mathbf{R}_X}\|t$ and $t_Y = \|\boldsymbol{\omega}_{\mathbf{R}_Y}\|t$. The coefficients c_i can be obtained by straightforward calculation. When replacing $\tan t_X$ and $\tan t_Y$ by x and y, respectively so that $\cos t_X = \frac{1-x^2}{x^2+1}$, $\cos t_Y = \frac{1-y^2}{y^2+1}$, $\sin t_X = \frac{2x}{x^2+1}$, $\sin t_Y = \frac{2y}{y^2+1}$, the above line search reduces to a multivariate polynomial root-finding problem for which exact solutions are available [28].

The use of the exact stepsize formula reduces the number of iterations required for convergence. However, there are several associated drawbacks from both the theoretical and computational perspective. First, the function value along the geodesic between $(\mathbf{R}_{X_k}, \mathbf{R}_{Y_k})$ and $(\mathbf{R}_{X_{k+1}}, \mathbf{R}_{Y_{k+1}})$ may not be strictly descending, and the updated point may end up in a different region of attraction. These features can violate the assumptions of the optimal Bayesian stopping rules which are essential in typical stochastic global optimization algorithms [21]. Bayesian stopping rules are derived under the assumption that the entire path generated from the local search is contained in one region of attraction, and the function value over each path segment generated in each update of the local search is monotonically decreasing. Second, the computational costs involved in the calculation of the exact stepsize can actually increase the total computation time. It is widely accepted in optimization that as a matter of practice, it is far better to spend time and resources in computing the search direction and to calculate the stepsize in a rapid manner so that they ensure the objective function decreases.

For these reasons, we present a stepsize estimate that is computationally more efficient and always guarantees a strictly descending function value. The estimate is based on the fact that for a twice-differentiable function $f(t)$, if $f'(0) > 0$ and $|f''(t)| \leq c$ for all t, then $f'(t) \geq 0$ in the interval $-\frac{f'(0)}{c} \leq t \leq \frac{f'(0)}{c}$. Conversely, if $f'(0) < 0$ and $|f''(t)| \leq c$ for all t, then $f'(t) \leq 0$ in the interval $\frac{f'(0)}{c} \leq t \leq -\frac{f'(0)}{c}$. From the above equation, the stepsize estimate $t^* = -\frac{f'(0)}{c}$ ensures that $f(t^*) <$

$f(0)$.

Based on the derivations given in Appendix A.4, our stepsize estimate is

$$t^* = -\frac{\phi'(0)}{c} \tag{3.3.14}$$

where $\phi'(0) = \sum_{i=1}^{18} \lambda_i tr(\mathbf{P}_i \mathbf{R}_{X_k} + \mathbf{Q}_i \mathbf{R}_{Y_k}) tr(\mathbf{P}_i \mathbf{R}_{X_k}[\boldsymbol{\omega}_{\mathbf{R}_X}] + \mathbf{Q}_i \mathbf{R}_{Y_k}[\boldsymbol{\omega}_{\mathbf{R}_Y}]) + tr(\mathbf{P}_0 \mathbf{R}_{X_k}[\boldsymbol{\omega}_{\mathbf{R}_X}] + \mathbf{Q}_0 \mathbf{R}_{Y_k}[\boldsymbol{\omega}_{\mathbf{R}_Y}])$ and $c = |\lambda|_{max} \left(\|[\boldsymbol{\omega}_{\mathbf{R}_X}]\|^2 + \|[\boldsymbol{\omega}_{\mathbf{R}_Y}]\|^2 \right) + \sqrt{6} |\lambda|_{max} \sqrt{\|[\boldsymbol{\omega}_{\mathbf{R}_X}]^2\|^2 + \|[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2\|^2} + \sqrt{3} \left(\|\mathbf{P}_0 \mathbf{R}_{X_k}[\boldsymbol{\omega}_{\mathbf{R}_X}]^2\| + \|\mathbf{Q}_0 \mathbf{Y}_k[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2\| \right)$. Then, $(\mathbf{R}_X, \mathbf{R}_Y)$ is updated by

$$\mathbf{R}_{X_{k+1}} = \mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}] t^*} \tag{3.3.15}$$

$$\mathbf{R}_{Y_{k+1}} = \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}] t^*}. \tag{3.3.16}$$

3.3.4 Summary of Local Search Algorithm

The proposed local search algorithm is summarized in Algorithm 1. Like all descent algorithms, the proposed algorithm consists of direction finding and line searching step. For the geometric steepest descent method the search direction is given by

$$\begin{bmatrix} \boldsymbol{\omega}_{\mathbf{R}_X} \\ \boldsymbol{\omega}_{\mathbf{R}_Y} \end{bmatrix} = -\nabla J, \tag{3.3.17}$$

while for Newton's method

$$\begin{bmatrix} \boldsymbol{\omega}_{\mathbf{R}_X} \\ \boldsymbol{\omega}_{\mathbf{R}_Y} \end{bmatrix} = -[\nabla^2 J]^{-1} \nabla J, \tag{3.3.18}$$

where ∇J and $\nabla^2 J$ are as given in Equations (A.3.3) and (A.3.4) in Appendix A.3.

Algorithm 1: Local Search Algorithm

initialization: set $k = 0$ and compute the initial guess $(\mathbf{R}_{X_0}, \mathbf{R}_{Y_0})$ by using Equations (3.3.10) and (3.3.11);
while *local convergence criterion is not satisfied* **do**
    find $\boldsymbol{\omega}_{\mathbf{R}_X}$ and $\boldsymbol{\omega}_{\mathbf{R}_Y}$ by using Equation (3.3.17) or (3.3.18);
    compute the stepsize t^* by using Equation (3.3.14);
    compute $(\mathbf{R}_{X_{k+1}}, \mathbf{R}_{Y_{k+1}})$ by using Equations (3.3.15) and (3.3.16);
    $k \leftarrow k + 1$;
return the minimizer $(\mathbf{R}_X^*, \mathbf{R}_Y^*)$

3.4 Stochastic Global Optimization

In this section we propose a two-phase stochastic global optimization method for our nonlinear objective function (3.3.8). There are many variations of this method as described in [21], [22] but the main features are the same: (i) generate uniform samples on the search space S. (ii) determine whether or not to apply local search to each sample. (iii) apply local search to samples selected in (ii). (iv) add newly discovered local minimizer $\mathbf{x}^*$ to a set of local minimizers $\mathbb{X}^*$ (which is initially empty) and assign samples to their minimizers. (v) check the optimal Bayesian stopping criterion. (vi) Stop or go back to (i).

3.4.1 Uniform Random Sampling on $SO(3)$

There are several ways to generate a random sequence of rotations uniformly. The main difficulties are in the choice of a convenient parametrization for $SO(3)$, and sampling from the parameter space in a way such that the resulting samples on

$SO(3)$ are uniform [29]. The traditional way of performing uniform random sampling on $SO(3)$ is the subgroup algorithm presented in [30]. The group of orthogonal matrices $O(n)$ contains $O(n-1)$ as its subgroup. By randomly sampling an element of $O(n-1)$ and coset representatives for $O(n-1)$ in $O(n)$, it generates a uniform random sample of $O(n)$.

Algorithm 2: Generating Uniform Random Samples on $SO(3)$

1 Generate θ uniformly on $[0, 2\pi]$;

2 Generate $\mathbf{v} \in \mathbb{R}^3$ uniformly on $\mathbb{S}^2$ (2-dimensional sphere);

3 Set $\mathbf{x} = (\mathbf{e}_1 - \mathbf{v})/\|\mathbf{e}_1 - \mathbf{v}\|$ where $\mathbf{e}_1 = [\ 1 \quad 0 \quad 0\ ]^T$;

4 Generate a sample $\mathbf{R} \in SO(3)$ by

$$\mathbf{R} = \left(\mathbf{I}_3 - 2\mathbf{x}\mathbf{x}^T\right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & -\sin\theta & -\cos\theta \end{bmatrix}. \tag{3.4.19}$$

3.4.2 Resampling for Local Search

After generating the random samples, there is a resampling step before applying the local search algorithm in Algorithm 1 to each sample. There exist many variations of the stochastic global optimization with their own resampling strategies. Most of the variations focus on reducing the number of attempts at local search. For example, clustering methods evaluate function values of samples and the distances between samples to avoid unnecessary multiple attempts at local search in the same region of attraction. They assign some samples to their minimizers without applying local search. However, though they can save computation time, the quality of the solution is generally not dependent on the selection of the variation.

For this reason, we do not introduce any specific variation here.

3.4.3 Optimal Bayesian Stopping Rules

Two possible stopping rules are as follows:

(i) Using the posterior expectation of the number of local minima:

$$\frac{w(N-1)}{N-w-2} < w + \epsilon. \tag{3.4.20}$$

(ii) Using the posterior expected relative size of the non-observed regions of attraction:

$$\frac{w(w+1)}{N(N-1)} < \delta. \tag{3.4.21}$$

Here, w is the number of local minima discovered and N is the number of samples assigned to the minimizers. ϵ and δ are the stopping criteria. Note that N is not necessarily the same as the total number of attempts at local search. Some samples can be assigned to a nearby minimizer by checking the location and function value according to the resampling strategy.

To make reliable use of (3.4.20) and (3.4.21), one should ensure that the search space S is finite and the local search procedure is strictly descending and completely contained in S. We have already resolved the finite search space issue by reducing the search space to $SO(3) \times SO(3)$. The strictly descending stepsize is also addressed in the previous section with associated geometric local search algorithm.

3.4.4 Summary of Stochastic Global Optimization Algorithm

A stochastic global optimization algorithm using the local search algorithm is presented in Algorithm 3.

Algorithm 3: Stochastic Global Optimization Algorithm

1 initialization: set $N = 0$ and $\mathbb{X}^* = \{\}$;

2 uniform random sampling on $SO(3)$: generate n random samples of $(\mathbf{R}_X, \mathbf{R}_Y) \in SO(3) \times SO(3)$;

3 **while** *local convergence criterion is not satisfied* **do**

4 apply local search described in Algorithm 1 to the samples generated in Step **2**. (*Note: fewer than n local search calls are required if the reduction schemes presented in [21], [22] are applied.*);

5 assign k samples to corresponding local minimizer and insert newly discovered minimizer $(\mathbf{R}_X^*, \mathbf{R}_Y^*)$ into $\mathbb{X}^*$.;

6 $N \leftarrow N + k$;

7 return the best minimizer $(\mathbf{R}_X^*, \mathbf{R}_Y^*)$ in $\mathbb{X}^*$

3.5 Simulations

In this section, we compare the performances of our geometric algorithms against the state-of-the-art "local unit quaternion-based" nonlinear optimization [4] using synthetic data.

3.5.1 Synthetic Data

We generate synthetic data to simulate an aerial vehicle with infrared markers and onboard camera in Figure 3.1(b). Using these synthetic data, we present experimental results for three different optimization methods: (i) stochastic global geometric algorithm, (ii) local geometric algorithm, and (iii) local unit quaternion-based nonlinear optimization.

- For the local geometric approach, the initial guess $(\mathbf{R}_{X_0}, \mathbf{R}_{Y_0}) \in SO(3) \times SO(3)$ can be obtained by using the Equations (3.3.10) and (3.3.11). In these experiments, we set ζ in Equation (3.3.7) to 1.

- We randomly set a pair of ground truth rotations $(\mathbf{R}_{X,true}, \mathbf{R}_{Y,true}) \in SO(3) \times SO(3)$ by using the uniform random sampling method described in the previous section 3.4.1. Two true translations $(\mathbf{p}_{X,true}, \mathbf{p}_{Y,true}) \in \mathbb{R}^3 \times \mathbb{R}^3$ are randomly set. Taking into account the relative scale between the aerial vehicle and the workspace, we normalize $(\mathbf{p}_{X,true}, \mathbf{p}_{Y,true})$ differently as $\|\mathbf{p}_{X,true}\| = 0.1$ and $\|\mathbf{p}_{Y,true}\| = 1$.

- With the above pair of rotations and translations, we construct true values of $(\mathbf{X}_{true}, \mathbf{Y}_{true}) \in SE(3) \times SE(3)$. We now generate a set of N random pose measurements, $\{\mathbf{A}_i\}_{i=1,...N} \in SE(3)$, where the number of measurements $N \in [12, 40]$ is also randomly generated. To ensure realistic trajectories for the aerial vehicle within the limited workspace, the translation component of each $\mathbf{A}_i$ is confined to lie on a sphere with varying radius. The other set of pose measurements $\{\mathbf{B}_i\}_{i=1,...N}$ are uniquely determined as $\mathbf{B}_i = \mathbf{Y}_{true}^T \mathbf{A}_i \mathbf{X}_{true}$.

Noisy data sets are then generated by multiplying the rotations $(\mathbf{R}_{A_i}, \mathbf{R}_{B_i})$ of the trajectories by random noisy rotation matrices $(\exp([\mathbf{v}_A]), \exp([\mathbf{v}_B]))$ as follows: $\mathbf{R}_{A_i} \leftarrow \mathbf{R}_{A_i} \exp([\mathbf{v}_A])$ and $\mathbf{R}_{B_i} \leftarrow \mathbf{R}_{B_i} \exp([v_B])$ where $\mathbf{v}_A, \mathbf{v}_B \in \mathbb{R}^3 \sim \mathcal{N}(0, (\gamma\pi)^2\mathbf{I})$ are Gaussian noises and $\gamma \in \mathbb{R}$ is the noise level. The translation $(\mathbf{p}_{A_i}, \mathbf{p}_{B_i})$ are also corrupted by $\mathbf{p}_{A_i,x} \leftarrow \mathbf{p}_{A_i,x} + \mathbf{p}_{A_i,x}\mathbf{w}_A$ and $\mathbf{p}_{B_i,x} \leftarrow \mathbf{p}_{B_i,x} + \mathbf{p}_{A_i,x}\mathbf{w}_B$ where $\mathbf{w}_A, \mathbf{w}_B \in \mathbb{R}^3 \sim \mathcal{N}(0, \gamma^2\mathbf{I})$. The y and z components of translation vectors $\mathbf{p}_{A_i}$ and $\mathbf{p}_{B_i}$ are corrupted in the same manner with the x component. For each noise level, $1,000$ trials of independent experiments are performed

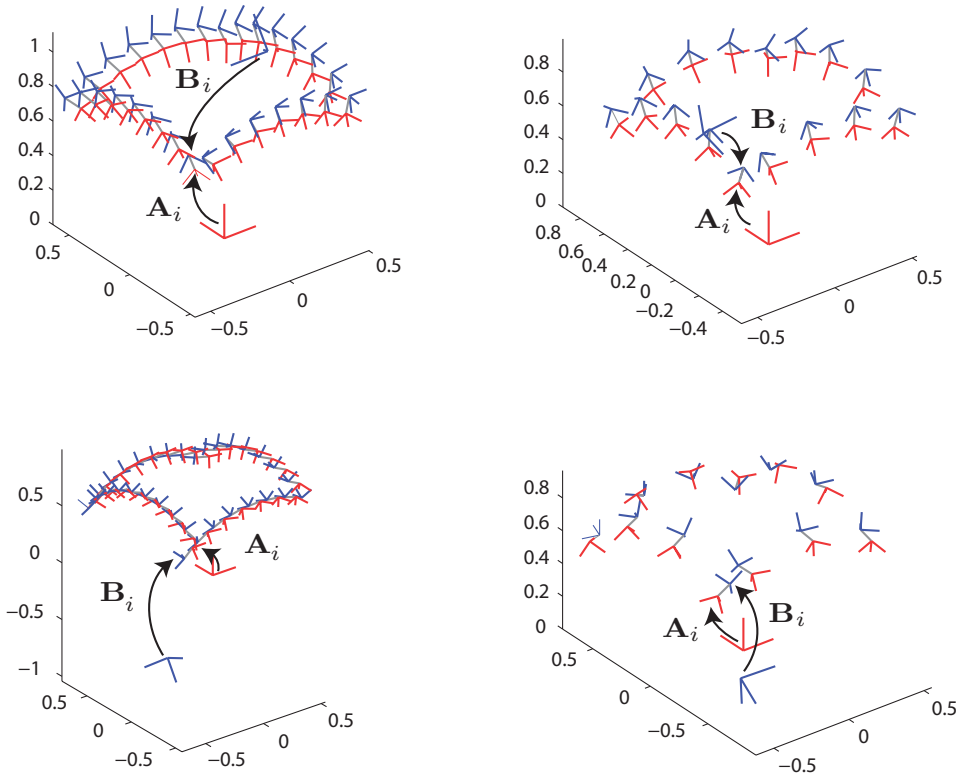
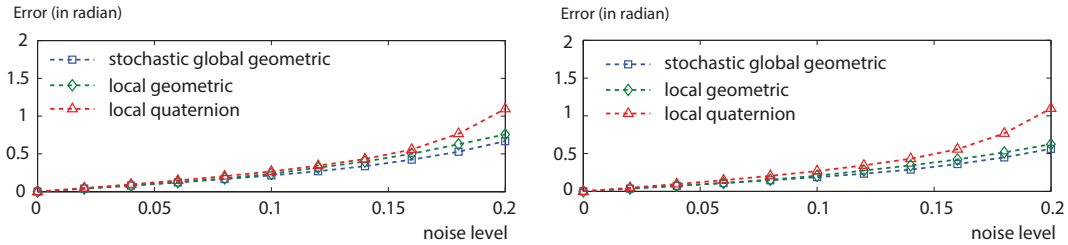
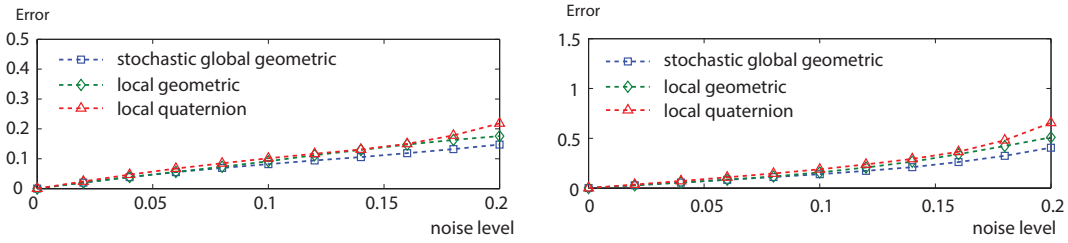


Figure 3.2: Synthetically-generated continuous trajectories of pose data $(\mathbf{A}_i, \mathbf{B}_i)$ to simulate a real aerial vehicle with infrared markers and onboard camera in Figure 3.1(b).



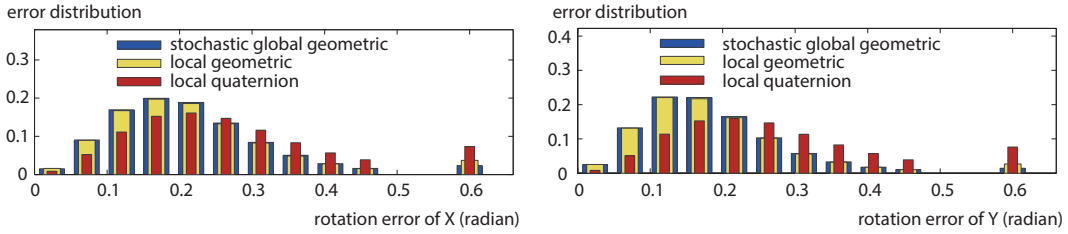
(a) rotation error of X (b) rotation error of Y

Figure 3.3: Geodesic angle errors with increasing noise level.



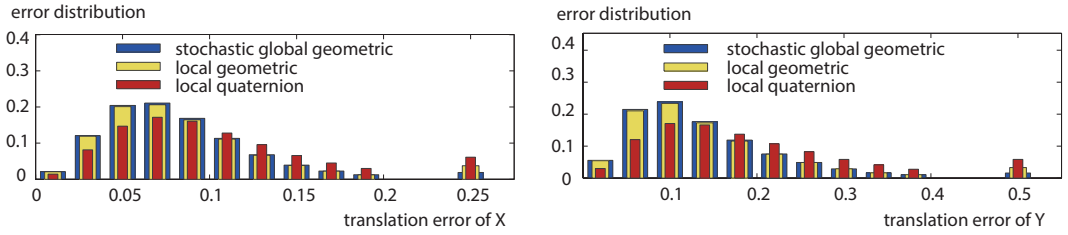
(a) translation error of X (b) translation error of Y

Figure 3.4: Translation errors with increasing noise level.



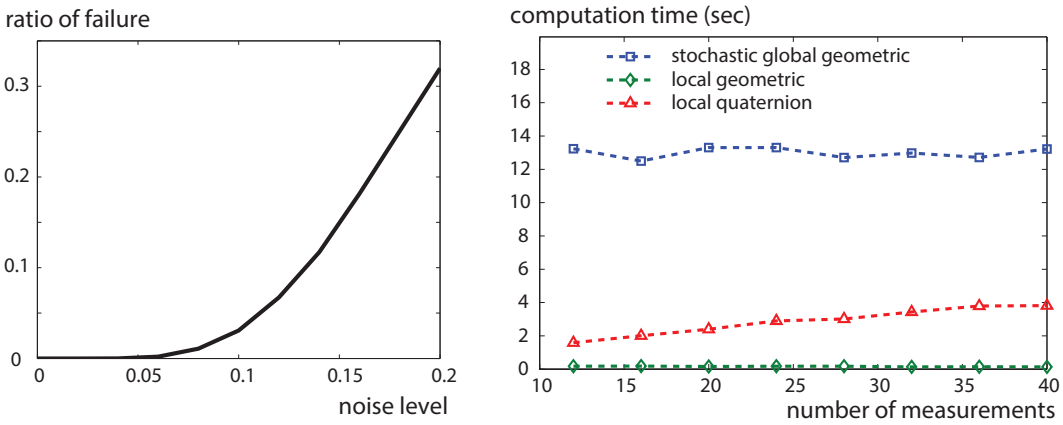
(a) distribution of rotation error of X (b) distribution of rotation error of X

Figure 3.5: Distribution of geodesic angle errors when $\gamma = 0.1$.



(a) distribution of translation error of X (b) distribution of translation error of Y

Figure 3.6: Distribution of translation errors when $\gamma = 0.1$.



(a) ratio of deviation (b) average computation times

Figure 3.7: (a) ratio of deviations from a stochastic global solution in the local geometric method, (b) average computation times with respect to the number of measurements when $\gamma = 0.1$ 100 experimental trials are averaged for each number of measurements.

to regress the random effect in a single experiment. We perform this experiment 100 times with different true values $(\mathbf{X}_{true}, \mathbf{Y}_{true}) \in SE(3) \times SE(3)$ and measurements $\{\mathbf{A}_i, \mathbf{B}_i\}_{i=1,\dots N} \in SE(3) \times SE(3)$ as shown in Figure 3.2. As a result, we ultimately perform $100 \times 1,000$ experiments with 100 sets of independent pose trajectories and $1,000$ different noise distributions at each noise level.

We evaluate our stochastic global geometric algorithm and some local algorithms by comparing the accuracy of the estimates $(\mathbf{X}_{est}, \mathbf{Y}_{est})$ with the synthetically generated ground truth values $(\mathbf{X}_{true}, \mathbf{Y}_{true})$. Figures 3.3 and 3.4 show the average rotation errors (in terms of geodesic angle) and translation errors, respectively, for noise levels varying within the range $\gamma \in [0, 0.2]$. With increasing noise levels γ, the stochastic global geometric algorithm shows superior performance over both the local quaternion-based method and the local geometric method. As the noise level increases, the error of the local geometric algorithm deviates from that of the stochastic global geometric algorithm. This can be attributed to the local solution tending to converge to another local minimizer with increasing noise. The ratio of deviations from the stochastic global solution in the local geometric method is shown in Figure 3.7(a). Note that the ratio of deviations from the stochastic global solution increases with noise, since large noise levels increase the number of local minima that attract the solution to the problem at each iteration.

Figures 3.5 and 3.6 show error distributions in the rotations and translations over $100 \times 1,000$ experimental trials, with the noise level γ set to 0.1. The error distribution for the stochastic global geometric method tends to be much more sharply concentrated at small error values when compared to the results obtained for the local geometric and local quaternion-based methods. For problems in which large noise levels create numerous local minima, our stochastic global optimization

method outperforms local optimization methods in terms of accuracy and robustness.

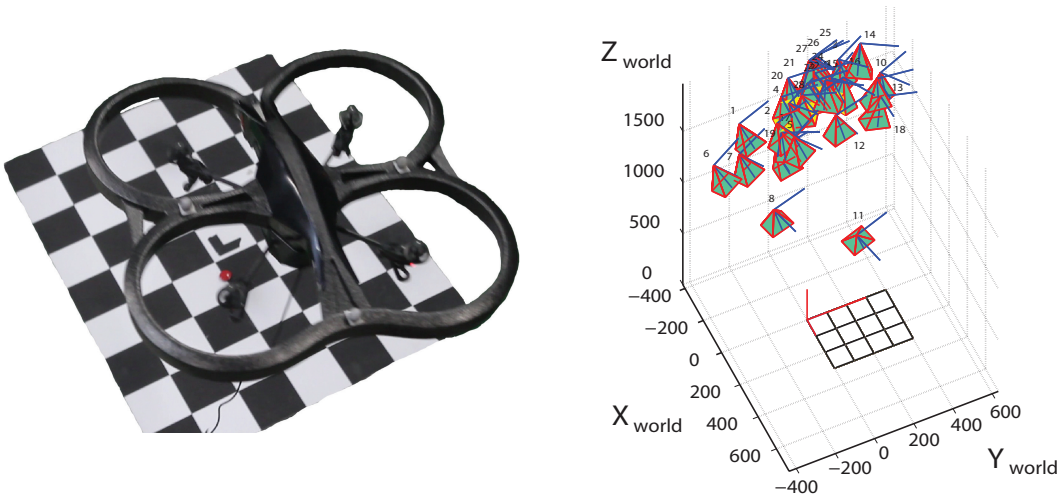
Figure 3.7(b) shows average computation times of the algorithms with respect to the number of measurement used in the calibration. 100 experimental trials are averaged for each number of measurements, with the noise level γ set to 0.1. The computation times of our global and local algorithms are invariant to the number of measurements since the time complexity of evaluating the objective function has been reduced to $O(1)$ as stated in Section 3.3.1. The average computation times of our global and local algorithms are 0.16 sec. and 12.99 sec., respectively. On the other hand, the computation time of the quaternion-based method depends on the number of measurements used in the calibration. It increases from 1.59 sec. to 3.81 sec. as the number of measurements increases from 12 to 40.

3.6 Applications

3.6.1 Camera-Marker Calibration for Unmanned Aerial Vehicle

We present experimental results of our calibration algorithm using real data obtained from actual multiple networked infrared (IR) cameras (OptiTrack$^{\text{TM}}$) and a color camera on a commercial UAV (Parrot AR.Drone$^{\text{TM}}$ 2.0) as shown in Figure 3.8(a).

Though the AR.Drone is equipped with two small cameras, we attach an additional wired color camera (Logitech$^{\text{TM}}$ webcam C905) onto the AR.Drone to support time synchronization with the OptiTrack IR camera system via a host computer. Whenever the webcam captures planar checkerboard images on the ground, the pose data A_i of the reflective IR markers (illustrated in Figure 3.1(b)) are recorded. The webcam pose data B_i with respect to the coordinate frame of the



(a) UAV with IR markers and a camera (b) Trajectory of camera on the UAV

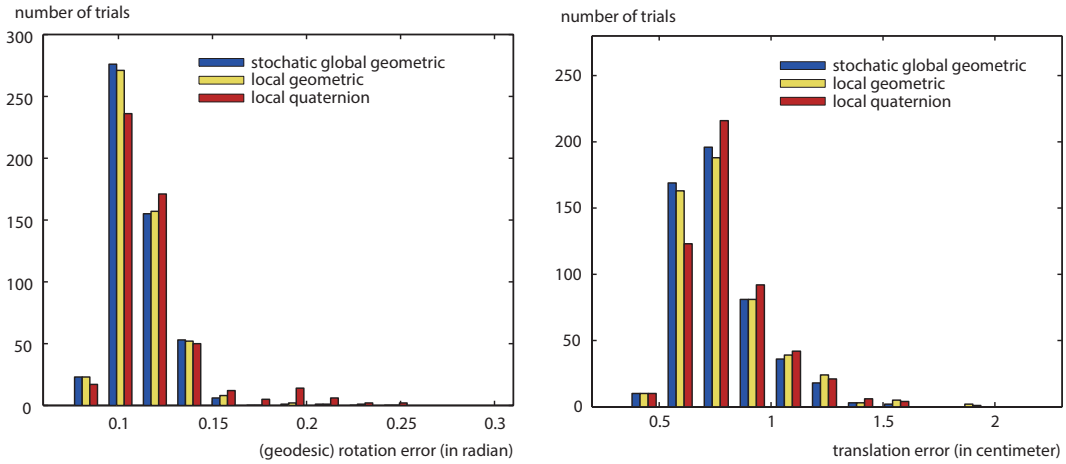
Figure 3.8: Screen shot of our experimental setup: a UAV equipped with IR reflective markers and a color camera.

checkerboard are easily computed by using a standard method for camera calibration [31]. Figure 3.8(b) shows the trajectory of the webcam on a UAV for this experiment.

Table 3.1: Average estimation errors, $\frac{1}{N_e}\sum_{k=1}^{N_e} E_{geod}$ (in radian) and $\frac{1}{N_e}\sum_{k=1}^{N_e} E_t$ (in centimeter); $N_0 = 7, N_t = 28, N_e = 515$

	stochastic global geometric	local geometric	local quaternion
$\frac{1}{N_e}\sum_{k=1}^{N_e} E_{geod}$	**0.1108**	0.1115	0.1167
$\frac{1}{N_e}\sum_{k=1}^{N_e} E_t$	**0.7725**	0.7902	0.8050

In general, when dealing with real data, the ground truth pair $(\mathbf{X}, \mathbf{Y})$ is unknown. Therefore, we employ a two-step verification procedure for our experiments, involving an estimation step and a validation step [32]. In the estimation step,



(a) histograms of rotation errors

(b) histograms of translation errors

Figure 3.9: histogram of rotation and translation errors

we randomly select N_0 measurement pairs from a total of N_t measurement pairs $(\mathbf{A}_i, \mathbf{B}_i)$, $(i = 1 \dots N_t)$. In each trial of random selection of measurement pairs, we compute the optimal estimate $(\mathbf{X}_{est}, \mathbf{Y}_{est})$ using only a randomly selected number N_0 of measurement pairs. For the validation step, we compute the estimation error by using the remaining $N_r = (N_t - N_0)$ data pairs excluding the previous N_0 randomly selected data pairs. We perform this two-step verification step N_e times.

The geodesic rotation error E_{geod} and the translation error E_t for each two-step verification trial are defined as

$$E_{geod} = \frac{1}{N_r} \sum_{j=1}^{N_r} \| \log(\mathbf{R}_{A_j} \mathbf{R}_X (\mathbf{R}_Y \mathbf{R}_{B_j})^{-1}) \| \tag{3.6.22}$$

$$E_t = \frac{1}{N_r} \sum_{j=1}^{N_r} \| \mathbf{R}_{A_j} \mathbf{p}_X + \mathbf{p}_{A_j} - \mathbf{R}_Y \mathbf{p}_{B_j} - \mathbf{p}_Y \|. \tag{3.6.23}$$

The histograms in Figures 3.9(a) and (b) show the rotation and translation estimation errors for several methods. For these experiments, $N_0 = 7$, $N_t = 28$ and

Figure 3.10: Humanoid robot with built-in cameras on its eyes.

$N_e = 515$. Note that our stochastic global geometric method with random initial guess outperforms the existing quaternion-based nonlinear optimization method with a closed-form initial guess as shown in Table 3.1.

3.6.2 Head-Eye Calibration for Humanoid Robot

As the second application, we conduct the experiment of head-eye calibration for a humanoid robot. The humanoid robot has a 2-DOF neck joint as shown in Figure 3.11. An off-the-shelf stereo camera (a Point Grey bumblebee camera with 320×240 pixel resolutions) system is rigidly mounted on the head of the robot to emulate two eyes. As shown in Figure 3.12, the robot can move its head by controlling two motors in the neck joint and momentarily stopped to take a picture of a remote checkerboard. Let $\theta_{1,i}$ and $\theta_{2,i}$ be the joint angles about the axes of yaw and pitch directions at the i^{th} time step (see Figure 3.11). In our experiment,

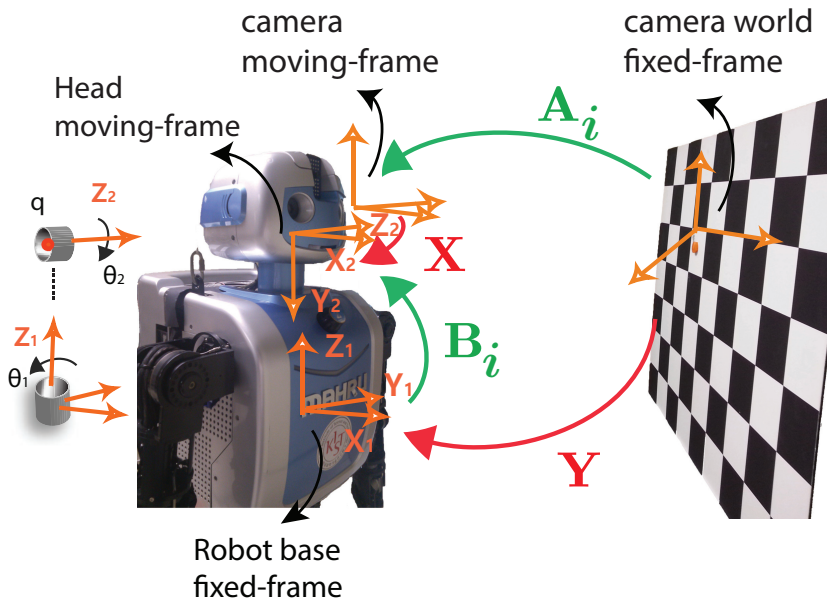


Figure 3.11: Reference frames for head-eye calibration.

the ranges of joint variables are given by $\theta_{1,i} \in [-12.5°, 12.5°]$ and $\theta_{2,i} \in [0°, 12.5°]$. From the measurements of joint variables $(\theta_{1,i}, \theta_{2,i})$, we can compute forward kinematics of the robot head by using the product of exponentials (POE) formula [25]:

$$\mathbf{B}_i = f(\theta_{1,i}, \theta_{2,i}) = \exp(\mathbf{C}_1\theta_{1,i}) \exp(\mathbf{C}_2\theta_{2,i})\mathbf{M} \in SE(3)$$

where $\mathbf{C}_k = \begin{pmatrix} [\mathbf{n}_k] & \mathbf{v}_k \\ \mathbf{0} & 0 \end{pmatrix} \in se(3)$, $(k = 1, 2)$. In our configuration, $\mathbf{n}_1 = (0, 0, 1)^T$, $\mathbf{v}_1 = (0, 0, 0)^T$, $\mathbf{n}_2 = (0, 1, 0)^T$, $\mathbf{v}_2 = (-0.4642, 0, 0)^T$ and $\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0.12 \\ 0 & -1 & 0 & 0.3542 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in SE(3)$. Here $\mathbf{M}$ represents the configuration of the robot's head frame relative to the base frame when the robot is at its home position. The physical length units of the translational part of $\mathbf{M}$ are meters.

A checkerboard pattern with each checker size 100×100 mm is attached to a

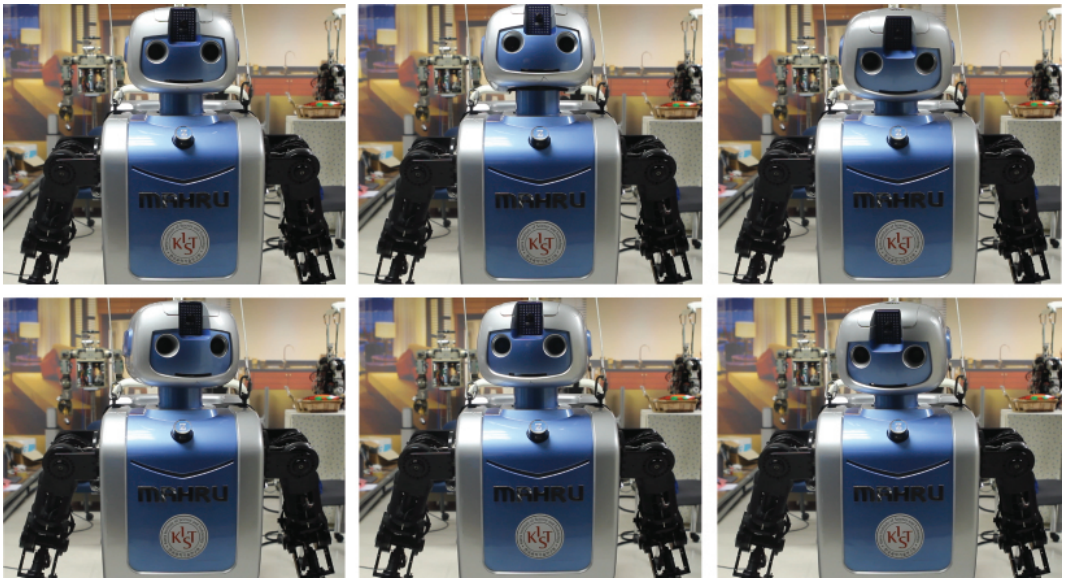
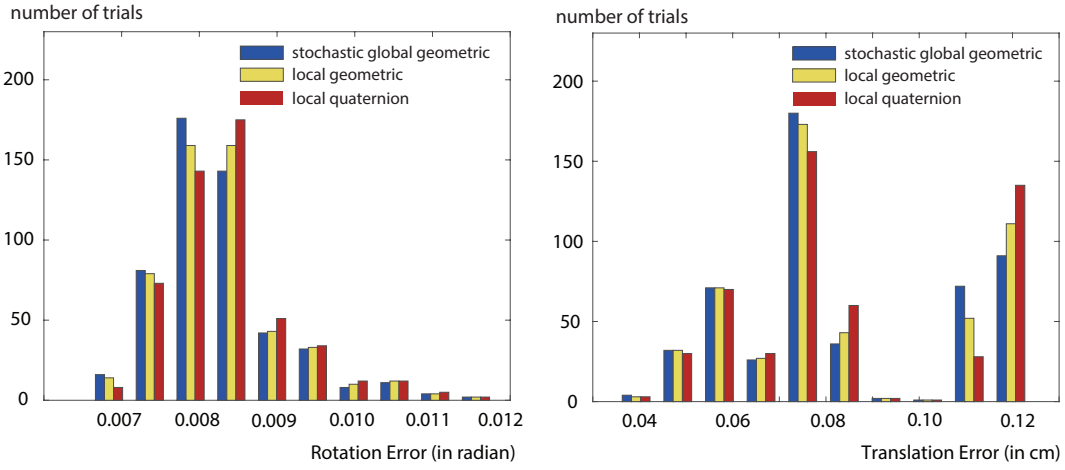


Figure 3.12: Random head-eye movement of humanoid robot.



(a) histogram of rotation errors

(b) histogram of translation errors

Figure 3.13: Histograms of calibration errors.

Table 3.2: Average estimation errors, $\frac{1}{N_e}\sum_{k=1}^{N_e} E_{geod}$ (in radian) and $\frac{1}{N_e}\sum_{k=1}^{N_e} E_t$ (in cm); $N_0 = 7, N_t = 28, N_e = 515$

	stochastic global geometric	local geometric	local quaternion
$\frac{1}{N_e}\sum_{k=1}^{N_e} E_{geod}$	**0.00823**	0.00827	0.00834
$\frac{1}{N_e}\sum_{k=1}^{N_e} E_t$	**0.08483**	0.08699	0.08760

wall. After capturing 56 checkerboard images, the i^{th} camera pose, $\mathbf{B}_i$ corresponding to the robot left eye pose is obtained through a standard calibration method.

When dealing with real data, the ground truth $(\mathbf{X}, \mathbf{Y}) \in SE(3) \times SE(3)$ pair is unknown. Therefore, we adopt the two-step verification procedures: estimation step and validation step. In an estimation step, we randomly selected N_0 number of measurement pairs from a total of N_t measurement pairs (A_i, B_i), $(i = 1 \dots N_t)$. In each trial of random selection of measurement pairs, we computed the optimal estimate (X_{est}, Y_{est}) only using randomly selected N_0 number of measurement pairs. Then, as a validation step, we computed our estimation error by using the remaining $N_r = (N_t - N_0)$ data pairs without containing the previous N_0 randomly selected data pairs. We performed this two-step verification step N times.

Table 3.2 summarizes the results of estimation errors in rotation and translation using several methods. In our experiments, $N_0 = 20$, $N_t = 66$ and $N = 515$. Note that our globally optimal geometric method with random initial guesses is superior to the conventional quaternion-based nonlinear minimization method as shown in the Table 3.2.

3.6.3 Affine Registration for Improving the Accuracy of Eye Trackers

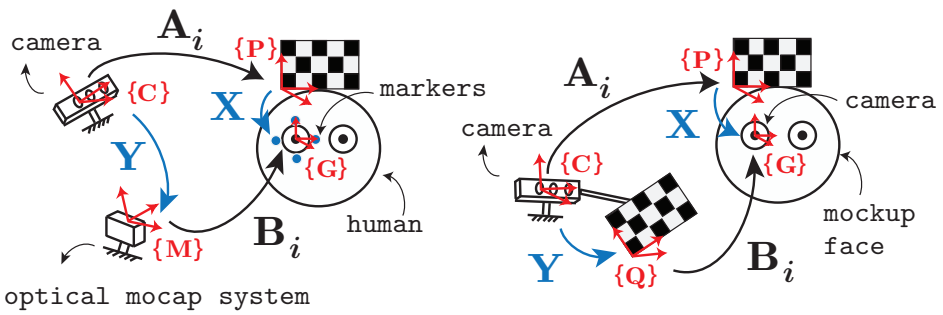
3.6.3.1 Motivation

Autostereoscopic display systems can provide 3-D images for viewers without requiring any special glasses [33, 34, 35]. In such systems, the information about viewers' 3-D eye positions relative to the display screen can provide a useful clue to reduce crosstalk and extend viewing zones [36, 37, 38]. In this regard, the information of viewers' 3-D eye positions can be an important ingredient for realistic autostereoscopic visualization.

In theory, 3-D eye positions can be easily obtained by using stereo triangulation if two calibrated cameras are available and a pair of salient 2-D feature points corresponding to the eye can be correctly detected by some methods [39, 40]. From epipolar geometry, the physical length of the baseline (i.e., the line connecting two camera centers) can give a useful information about the absolute scale of the observed environment. For this reason, a dedicated camera system consisting of a stereo high-resolution camera and some active infrared (IR) illumination devices is commonly used to track 3-D positions of pupils and estimate gaze directions [40, 41]. However, when considering that eye tracking sensors (typically cameras) are usually attached to display devices, the camera systems under active IR illumination are not suitable for some autostereoscopic display systems with large viewing distances between the display screen and viewers. This is because the working volume of the camera systems with active IR illumination devices for tracking small-sized pupils are mostly small. To resolve the issue of the small working volume inherent to these types of systems employing the camera and active IR illumination devices, some researchers use several additional devices such as a zoom-lens camera [42] or a pan-tilt unit [43] in order to track remote pupils.

Rather than directly tracking small-sized pupils by using high-resolution cameras with active IR illumination devices, there exist other methods for estimating 3-D eye positions by tracking boundary feature points of the eyes including eye corners with a monocular low-resolution color camera. However, when using a single low-resolution color camera, 2-D feature points around the eyes can be erroneously detected by feature extraction algorithms owing to illumination changes, cluttered background images, or non-frontal head poses relative to the cameras. Moreover, without reasonable assumptions and constraints, it is an ill-posed problem to obtain 3-D eye positions on the absolute scale by using only a monocular color camera [44]. Common approaches for resolving the issue of inferring the absolute scale of 3-D facial features by using a single 2-D facial image are data-driven, learning-based, and dependent on deformable 3-D face models [45, 46, 47]. In these approaches, the accuracy of the estimated 3-D eye positions from a single 2-D face image is susceptible to the variations of head poses related to the employed 3-D face models and face databases [48]. Although there are numerous methods [48, 49] for constructing sophisticated 3-D face models using a single camera, there exists almost no face database consisting of monocular 2-D face images annotated by the metric information about the corresponding ground-truths of 3-D eye positions. For this reason, existing methods for estimating 3-D eye positions on the absolute scale from monocular 2-D images may yield erroneous results. To improve the accuracy of eye trackers using a single color camera, some compensation methods are required.

In this thesis, we present a systematic compensation method for improving the accuracy of 3-D eye position trackers using a "monocular" low-resolution color camera by determining an optimal registration function for data fitting. The problem of improving the accuracy of an eye position tracker can be viewed as the



(a) markers around an eye (b) mockup face with a chceckerboard

Figure 3.14: Two different systems to obtain the ground-truths of 3-D eye positions

problem of fitting two sets of 3-D point data consisting of the inaccurate 3-D eye position estimates from the eye tracker with a single camera and the ground-truths of the corresponding 3D eye positions. To perform 3-D data fitting, we first need to find the ground-truths of 3-D eye positions and then determine a registration function that maps the 3-D eye position estimate onto the ground-truth of the corresponding 3-D eye position.

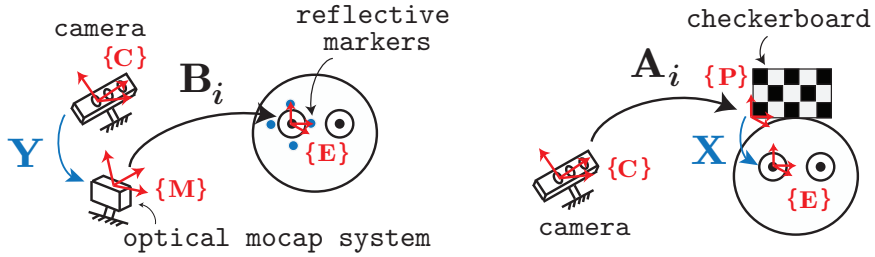
To obtain the ground-truths of 3-D eye positions with a single color camera, we propose two different types of systems as shown in Figure3.14(a) and (b), which combine an optical motion capture (mocap) system and checkerboards. From these setups, we can construct the system as the form of "two-frame sensor calibration" [3, 4, 1] to obtain the ground-truths of 3-D eye positions. As shown in Figure3.14 (a), several reflective markers around an eye are attached and a hat can be rigidly combined with a checkerboard on the head. Since the optical mocap system with multiple infrared cameras is generally expensive, we devise a relatively inexpensive apparatus as an alternative device to obtain the ground-truths of eye positions by using an additional camera and checkerboards (see Figure3.14 (b)).

After obtaining the ground-truths of 3-D eye positions by composing the systems as in Figure3.14 and using the algorithm for the two-frame sensor calibration, the next task is to find a registration function, which fits the inaccurate 3D eye position estimates into the ground-truth of 3-D eye positions. To confine the solution space and reduce overfitting side-effects, we assume that the registration function for fitting 3-D data has the form of an affine function, which can be derived from the first order approximation. By solving a least-squares optimization problem, we can determine the optimal affine function in an analytic form. Determination of the registration function can be regarded as the offline process for parameter estimation. With the registration function in an affine form, the 3-D eye positions estimated by the eye tracker using a single camera can be compensated in real-time. To demonstrate the generality and effectiveness of the proposed method regarding the approximation of the registration function in an affine form, we perform extensive experiments by setting up a system as illustrated in Figure3.14 (b).

3.6.4 Problem Statement

Let us first assume that a set of 3D facial features including eye corner features can be obtained at every time step by using a certain 3D face tracking algorithm with a monocular color camera. Let $\{C\}$ and $\{E\}$ denote the coordinate frames fixed to the monocular color camera and the eye, respectively. From 3D facial features at time step i, we can easily extract the 3D eye position $\boldsymbol{\xi}_i \in \mathbb{R}^3$, which represents the vector from the origin of $\{C\}$ to the origin of $\{E\}$ expressed in $\{C\}$. When $\boldsymbol{\xi}_i$ is not accurate enough, we need to adjust $\boldsymbol{\xi}_i$ to the correct value as a post-processing procedure.

A reasonable approach for correcting $\boldsymbol{\xi}_i$ may involve finding a certain registration function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that maps the inaccurate $\boldsymbol{\xi}_i$ into a more accurate



(a) using optical mocap system (b)using a checkerboard

Figure 3.15: Unknown coordinate transformations $\mathbf{X}, \mathbf{Y} \in SE(3)$ that are constant

value called the *ground truth*. To approximate f, we should first find the *unknown* ground-truth of the 3D eye position. Let $\mathbf{g}_i \in \mathbb{R}^3$ denote the unknown ground-truth of the 3D eye position corresponding to $\boldsymbol{\xi}_i$.

To obtain $\mathbf{g}_i$, one may consider some additional devices like an optical mocap system or a checkerboard (see Figure3.15(a) and (b)). However, in these cases, an important issue related to the unknown coordinate transformations may arise. To be more specific, let us fist consider an optical mocap system consisting of multiple networked optical IR cameras and reflective markers, which is commonly believed to provide highly accurate poses of a set of markers relative to the optical camera system. As shown in Figure3.15(a), several reflective markers are assumed to be put around the eye. In this situation, the optical mocap system can provide the pose of $\{E\}$ denoted by $\mathbf{B}_i = \begin{bmatrix} \mathbf{R}_{B_i} & \mathbf{p}_{B_i} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$ relative to the coordinate frame of the optical mocap system, $\{M\}$. Let $\mathbf{Y} \in SE(3)$ denote the unknown rigid body transformation of the frame $\{C\}$ relative to the frame $\{M\}$ and $\mathbf{e}_4 = (0, 0, 0, 1)^T$. The representation of $\mathbf{g}$ in *homogeneous coordinates* can be denoted as $\breve{\mathbf{g}}_i = (\mathbf{g}_i^T, 1)^T \in \mathbb{R}^4$ by appending a 1 to $\mathbf{g}$. If $\mathbf{Y}$ can be determined by some methods in advance, we can calculate $\breve{\mathbf{g}}_i$ from

$$\breve{\mathbf{g}}_i = \mathbf{Y}\mathbf{B}_i\mathbf{e}_4, \tag{3.6.24}$$

where the vector $\mathbf{YB}_i\mathbf{e}_4$ is the translational part of the pose $\mathbf{YB}_i$.

One may consider attaching a checkerboard on a head as shown in Figure3.15(b). Then, we can obtain

$$\breve{\mathbf{g}}_i = \mathbf{A}_i\mathbf{X}\mathbf{e}_4. \tag{3.6.25}$$

3.6.5 Method

In this section, we first present a method for obtaining the ground truth $\mathbf{g}_i$ of 3D eye position in Figure3.15(a) by determining the unknown constant pose $\mathbf{Y}$ (or $\mathbf{X}$) as shown in Figure3.15(a) (or (b)). Then, from given pairs of 3D points $(\boldsymbol{\xi}_i, \mathbf{g}_i)$, $(i = 1, \ldots, N)$ where N denotes the number of pose measurements, we will approximate the registration function f to the first order by solving a least-square optimization problem.

3.6.5.1 Ground Truths of Eye Positions

To determine $\mathbf{Y}$ (or $\mathbf{X}$) in Figure3.15(a) (or (b)), we set up a system as depicted in Figure3.14(a)(or (b)) by adding additional apparatuses to the original system. Let us first consider the system in Figure3.15(a) that uses a precise measurement device like an optical mocap system to obtain an accurate eye position. To obtain $\mathbf{g}_i$ from Equation(3.6.24), we should determine $\mathbf{Y}$ beforehand. The idea of attaching a checkerboard rigidly to the head as illustrated in Figure3.14(a) can provide a useful condition, which can give a clue to determine $\mathbf{Y}$ by collecting additional pose measurement $\mathbf{A}_i$. Here, $\mathbf{A}_i$ shown in Figure3.14(a) represents the pose of the checkerboard frame $\{P\}$ relative to the camera frame $\{C\}$ at time step i and it can be easily obtained by using a standard camera calibration method [31]. The pose of the eye coordinate frame $\{E\}$ defined by the reflective markers in Figure3.14(a)

relative to the camera frame $\{C\}$ can expressed as $\mathbf{A}_i\mathbf{X}$ or equally $\mathbf{Y}\mathbf{B}_i$. Note that $\mathbf{A}_i\mathbf{X}$ represents the equivalent rigid body transformation to $\mathbf{Y}\mathbf{B}_i$.

Let us now consider another system as illustrated in Figure3.15(b), where there is unknown rigid body transformation $\mathbf{X}$ between the checkerboard frame $\{P\}$ and the eye coordinate frame $\{E\}$. To obtain $\mathbf{g}_i$ from Equation(3.6.25), $\mathbf{X}$ should be identified in advance. To determine $\mathbf{X}$, we can construct the system as depicted in Figure3.14(b). In contrast to the system illustrated in Figure3.14(a) where a person is the subject of the experiments, the system in Figure3.14(b) employs a mockup face, of which eye is replaced by a webcam. As depicted in Figure3.14(b), two distinct checkerboards are rigidly attached to a camera on the ground and a mockup face, respectively. By using a standard camera calibration method, we can obtain pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$, $i = 1, \ldots, N$. Given noisy pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$, $\mathbf{X}$ and $\mathbf{Y}$ in Figure3.14(a) can be determined by minimizing

$$J(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{X} - \mathbf{Y}\mathbf{B}_i\|^2,$$

By using pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$ and the transformation $\mathbf{X}$ (or $\mathbf{Y}$), we can obtain the ground-truth $\mathbf{g}_i$ of 3D eye position in Figure3.14 from Equation(3.6.25) (or Equation(3.6.24)).

3.6.5.2 Affine Registration for Fitting Two Sets of 3-D Points Data

In the previous subsection, we have presented how to obtain $\mathbf{g}_i$ by constructing the system as shown in Figure3.14. We now present a method for approximating a registration function f that maps $\boldsymbol{\xi}_i$ into $\mathbf{g}_i$. Given pairs of 3D points $(\mathbf{g}_i, \boldsymbol{\xi}_i), (i = 1 \ldots N)$, the registration function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ can be written as $\mathbf{g}_i \approx f(\boldsymbol{\xi}_i) = f(\mathbf{c}) + \nabla f(\mathbf{c})(\boldsymbol{\xi}_i - \mathbf{c}) + \cdots$ where $\mathbf{c} \in \mathbb{R}^3$ is unknown. Here $f(\boldsymbol{\xi}_i)$ can be approximated by $f(\boldsymbol{\xi}_i) \approx h(\boldsymbol{\xi}_i) := f(\mathbf{c}) + \nabla f(\mathbf{c})(\boldsymbol{\xi}_i - \mathbf{c})$. The function h is the first order

approximation of f at a point $\mathbf{c}$ and can be rewritten as an affine function form,

$$h(\boldsymbol{\xi}_i) = \mathbf{S}\boldsymbol{\xi}_i + \mathbf{v}, \tag{3.6.26}$$

where $\mathbf{S} := \nabla f(\mathbf{c})$ is the 3×3 matrix and $\mathbf{v} := f(\mathbf{c}) - \nabla f(\mathbf{c})\mathbf{c}$ is the 3×1 column vector.

Given pairs of corresponding data $(\mathbf{g}_i, \boldsymbol{\xi}_i), (i = 1 \dots N)$, we can formulate the optimization problem by constructing the cost function J_0 as follows:

$$\text{minimize } J_0(\mathbf{S}, \mathbf{v}) = \sum_{i=1}^{N} \|\mathbf{g}_i - (\mathbf{S}\boldsymbol{\xi}_i + \mathbf{v})\|^2, \tag{3.6.27}$$

where $\mathbf{S}$ and $\mathbf{v}$ are the optimization variables. We now use the first-order necessary conditions to compute the optimal $\mathbf{S}$ and $\mathbf{v}$. From $\frac{\partial J_0(\mathbf{S},\mathbf{v})}{\partial \mathbf{v}} = \mathbf{0}$, we can obtain

$$\mathbf{v} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{g}_i - \mathbf{S}\boldsymbol{\xi}_i). \tag{3.6.28}$$

Substituting Equation(3.6.28) into (3.6.27) yields

$$J_0(\mathbf{S}) = \sum_{i=1}^{N} \|\tilde{\mathbf{g}}_i - \mathbf{S}\tilde{\boldsymbol{\xi}}_i\|^2, \tag{3.6.29}$$

where $\tilde{\boldsymbol{\xi}}_i := \boldsymbol{\xi}_i - \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\xi}_i$ and $\tilde{\mathbf{g}}_i := \mathbf{g}_i - \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}_i$. From $\frac{\partial J_0(\mathbf{S})}{\partial \mathbf{S}} = \mathbf{0}$ in Equation(3.6.29), we have

$$\mathbf{S} = (\sum_{i=1}^{N} \tilde{\mathbf{g}}_i \tilde{\boldsymbol{\xi}}_i^T)(\sum_{i=1}^{N} \tilde{\boldsymbol{\xi}}_i \tilde{\boldsymbol{\xi}}_i^T)^{-1}. \tag{3.6.30}$$

By using Equation(3.6.30), we can obtain the optimal $\mathbf{S}$. The optimal $\mathbf{v}$ can now be computed by substituting $\mathbf{S}$ into Equation(3.6.28). Thus far, we have obtained the optimal parameters, $\mathbf{S}$ and $\mathbf{v}$ in Equation(3.6.26) offline. For real time applications, we can correct $\boldsymbol{\xi}_i$ by using Equation(3.6.26).

3.6.6 Experimental Results using Real Data

Although our method considers a monocular color camera, it can be used to improve the accuracy of 3D eye positions estimated by any type of 3D face pose trackers using the RGB-D (red, green, blue and depth) camera system, of which extrinsic calibration parameters between an RGB camera and an IR camera are not sufficiently accurate. To verify this, a set of real data is obtained by using a 3D face tracker (Intel Realsense SDK) with the RGB-D camera, of which extrinsic calibration parameters are just used in a factory-calibrated setting. Since the factory-calibrated extrinsic parameters of the RGB-D camera are usually inaccurate, obviously the accuracy of 3D eye positions estimated by 3D face tracker using this camera system without customization will be inaccurate. The other set of real data will be collected by commercial face tracking software called FaceAPI$^{\text{TM}}$ using a monocular color camera.

In our experiments, we first set up the system as illustrated in Figure3.14(b). A webcam (Logitech C600) is installed on the right eye of a mockup face and a 7×9 checkerboard is rigidly attached to this mockup, of which checker square is of dimension 40 mm $\times$40 mm. A 7×9 checkerboard pattern image is displayed on the screen of the display device (Baytech Yamakasi QH2711 Black Label DP; maximum display pixel resolution is 2560×1440). The size of each checker on the screen is 36×36 mm^2.

3.6.6.1 The First Experiment: 3D Face Tracker using a Factory-Calibrated RGB-D Camera

As the first experiment, we mount the RGB-D camera (Intel RealSense$^{\text{TM}}$ F200) onto the display device. In this experiment, the extrinsic calibration parameters of

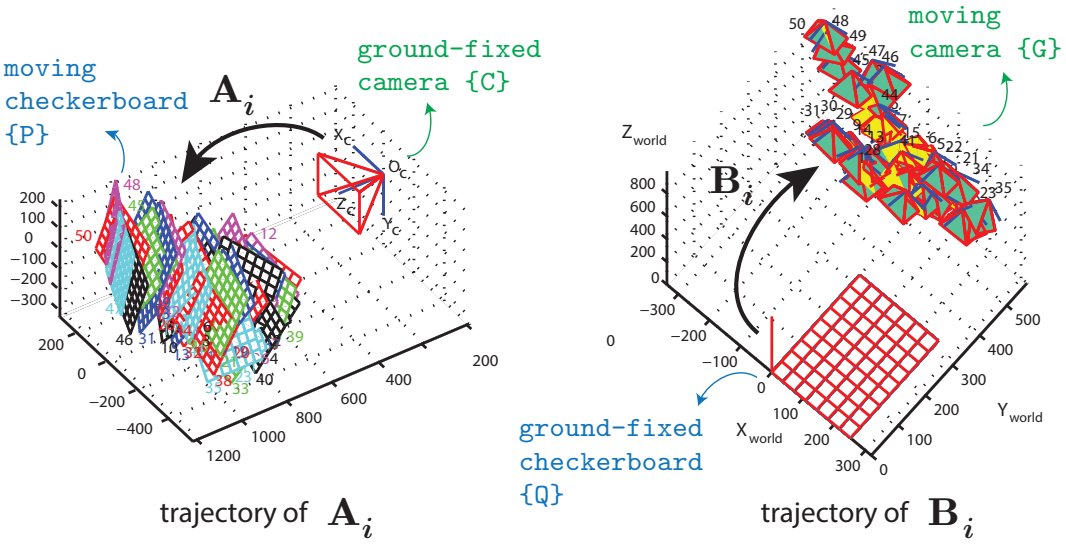
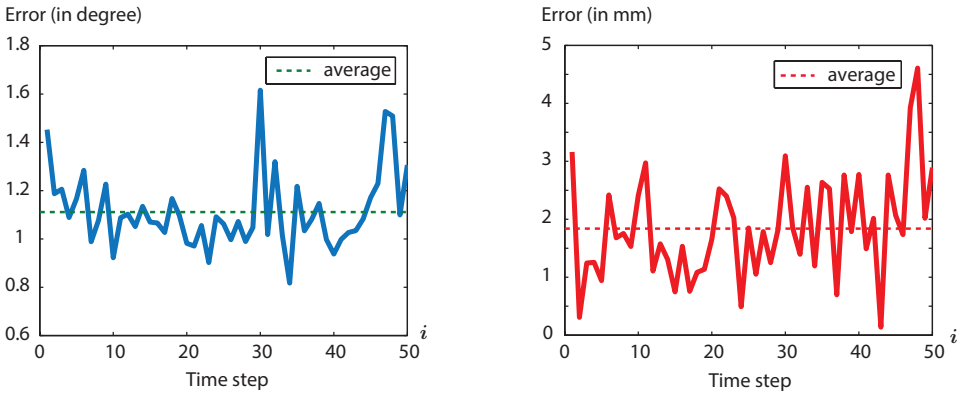


Figure 3.16: Trajectories of pose data for the two-frame sensor calibration

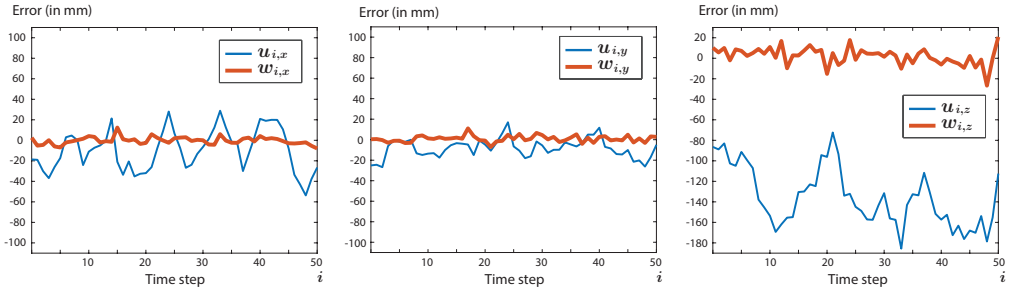
the RGB-D camera are just used in a factory-calibrated setting. The Intel RealSense$^{\text{TM}}$ SDK can provide 3D positions of facial feature points in metric unit with respect to its IR camera. By averaging the positions of four symmetric corner feature points around the right eye, we can obtain the estimate $\boldsymbol{\xi}_i$ $(i = 1 \ldots 50)$ of 3D eye positions from the eye tracker that is provided by the Intel RealSense$^{\text{TM}}$ SDK.

While keeping a mockup face stationary, we can obtain the images of the checkerboard on the display screen by using the webcam on the right eye of the mockup face. At the same time, the RGB-D camera on the display device captures the image of the checkerboard attached to the mockup face. We repeat this procedure 50 times with different poses of the mockup face. By using a standard camera calibration method [31], we can obtain pairs of pose data $(\mathbf{A}_i, \mathbf{B}_i)$ as well as $\boldsymbol{\xi}_i$, $(i = 1 \ldots 50)$. Figure3.16 shows the trajectory of $(\mathbf{A}_i, \mathbf{B}_i), (i = 1 \ldots 50)$.



(a) errors, $\boldsymbol{s}_{rot}^{(i)}$ (b) errors, $\boldsymbol{s}_{tran}^{(i)}$

Figure 3.17: The errors of two-frame sensor calibration



(a) errors at x-axis (b) errors at y-axis (c) errors at z-axis

Figure 3.18: When using a face tracker with a factory-calibrated RGB-D camera

Given the pose data $\mathbf{A}_i, \mathbf{B}_i$, $(i = 1, \ldots, 50)$, we can determine unknown constant poses $\mathbf{X}, \mathbf{Y}$ by using two-frame sensor calibration algorithm [1]. In our experiments, the checkerboard frame $\{Q\}$ in Figure3.14(b) can be considered as the screen coordinate frame. In this respect, we remark that $\mathbf{Y}$ represents a rigid body transformation of the display screen coordinate frame $\{Q\}$ relative to the camera coordinate frame $\{C\}$.

Let us denote the rotation error $s_{rot}^{(i)}$ and the translation error $s_{tran}^{(i)}$ $(i = 1, \ldots, 50)$

of the two-frame sensor calibration at time step i as

$$s_{rot}^{(i)} := \| \log(\mathbf{R}_{A_i}\mathbf{R}_X(\mathbf{R}_Y\mathbf{R}_{B_i})^{-1})\|$$
$$s_{tran}^{(i)} := \|\mathbf{R}_{A_i}\mathbf{p}_X + \mathbf{p}_{A_i} - \mathbf{R}_Y\mathbf{p}_{B_i} - \mathbf{p}_Y\|.$$

Figure3.17 shows the errors, $s_{rot}^{(i)}$ and $s_{tran}^{(i)}$, $(i = 1, \ldots, 50)$ representing the errors of the estimated rigid body transformations $(\mathbf{X}, \mathbf{Y})$ in terms of the rotational and translational parts. Let us define the averages of errors $s_{rot}^{(i)}$ and $s_{tran}^{(i)}$, $(i = 1 \ldots 50)$ as follows: $\bar{s}_{rot} := (1/50)\sum_{i=1}^{50} s_{rot}^{(i)}$ and $\bar{s}_{tran} := (1/50)\sum_{i=1}^{50} s_{tran}^{(i)}$. From Figure3.17, we can find that $\bar{s}_{rot} = 1.11^\circ$ and $\bar{s}_{tran} = 1.85$ mm.

Although two unknown poses $\mathbf{X}$ and $\mathbf{Y}$ have been determined, only one pose $\mathbf{X}$ (or $\mathbf{Y}$) is enough to compute $\mathbf{g}_i$ (see Equation(3.6.25) (or (3.6.24))). In our experiments, Equation(3.6.25) is used for computing $\mathbf{g}_i$. We now construct pairs of 3D points $(\mathbf{g}_i, \boldsymbol{\xi}_i), (i = 1 \ldots 50)$, from which the optimal parameters, $\mathbf{S}$ and $\mathbf{v}$ can be calculated by using Equations (3.6.28) and (3.6.30). Let us define $\mathbf{u}_i = (u_{i,x}, u_{i,y}, u_{i,y})^T \in \mathbb{R}^3$ and $\mathbf{w}_i = (w_{i,x}, w_{i,y}, w_{i,y})^T \in \mathbb{R}^3$ as follows:

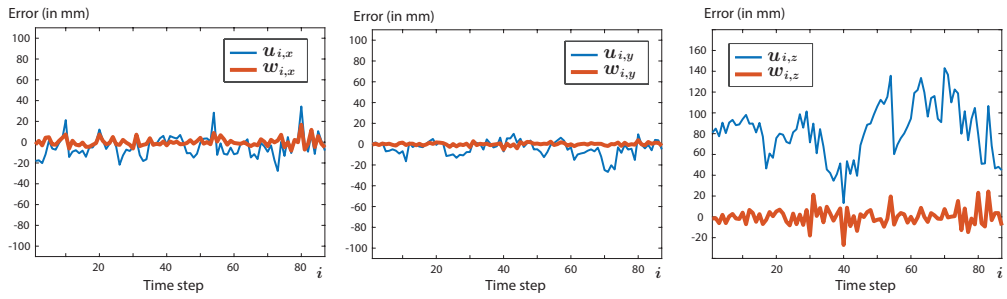
$$\mathbf{u}_i := \boldsymbol{\xi}_i - \mathbf{g}_i \tag{3.6.31}$$
$$\mathbf{w}_i := h(\boldsymbol{\xi}_i) - \mathbf{g}_i = \mathbf{S}\boldsymbol{\xi}_i + \mathbf{v} - \mathbf{g}_i. \tag{3.6.32}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the error vectors of $\boldsymbol{\xi}_i$ and $h(\boldsymbol{\xi}_i)$, respectively. As shown in Figure3.18, we can obtain much smaller errors after applying the registration function h.

3.6.6.2 The Second Experiment: 3D Face Tracker using a Monocular Color Camera

To validate the effectiveness of our method, of which main task is to determine two parameters $\mathbf{S}$ and $\mathbf{v}$ in Equation(3.6.26), we conduct another experiments using

a commercial 3D face tracking software, FaceAPI^{TM} with a monocular webcam (Logitech C905). By following the same procedure as the first experiment, we can collect data $\{(\mathbf{A}_i, \mathbf{B}_i, \boldsymbol{\xi}_i)|i = 1, \ldots, 87\}$ and then determine $(\mathbf{X}, \mathbf{Y})$ by using two-frame sensor calibration algorithm [1]. From Equations (3.6.31) and (3.6.32), we can compute the errors, $\mathbf{u}_i$ and $\mathbf{w}_i$ as shown in Figure3.19.



(a) errors at x-axis (b) errors at y-axis (c) errors at z-axis

Figure 3.19: When using FaceAPI^{TM} with monocular color camera

4

Geometric Unscented Kalman Filtering

4.1 Introduction

In this section, we first present an intrinsic (geometric) unscented Kalman filtering (UKF) algorithm evolving on a matrix Lie group in general setting. Then, as an application, we propose a geometric algorithm for estimating attitude (orientation) and slowly time-varying gyro bias in a coordinate-invariant way.

The first contribution of this research is to present an accurate UKF algorithm for estimating attitudes and gyro bias in a coordinate-invariant way. The novelty of this algorithm can be traced to the design of measurement function, which relates two different elements of matrix Lie groups, $SO(3) \times \mathbb{R}^3$ and $SO(3)$. Here $SO(3)$ denotes the rotation group and $\mathbb{R}^3$ represents a three-dimensional vector. In our formulation, the measurement has the form of $SO(3)$, which is given by the solution to Wahba's problem. Its merit is that the measurement noise covariance

can consider the constraint on two direction vectors and is also well-defined with a full rank.

As a second contribution, we provide an offline algorithm to determine the measurement noise covariance of the proposed UKF. By simply transforming the representation of the covariance from the IMU body-fixed frame into the ground-fixed frame, we find that the computation of the measurement noise covariance can be faster and more accurate when incorporated into the real-time attitude filter. Furthermore, we present an algorithm for determining the parameters in this covariance from measurements of gravity and geomagnetic field by using actual accelerometers and magnetometers.

4.2 Unscented Kalman Filtering on Matrix Lie Groups

Suppose $\mathcal{G}$ and $\mathcal{A}$ are the matrix Lie groups, of which Lie algebras are $\mathfrak{g}$ and $\mathfrak{a}$, respectively. Consider a state $\mathbf{X}_k \in \mathcal{G}$ and the dynamical system in the discrete-time

$$\mathbf{X}_{k+1} = \mathcal{F}(\mathbf{X}_k, \mathbf{n}_k) \in \mathcal{G} \tag{4.2.1}$$

$$\mathbf{Y}_{k+1} = \mathcal{H}(\mathbf{X}_{k+1}, \mathbf{w}_{k+1}) \in \mathcal{A}, \tag{4.2.2}$$

where $\mathbf{Y}_{k+1} \in \mathcal{A}$ denotes a measurement matrix. Here $\mathcal{F}, \mathcal{H}$ respectively represent the state transition and measurement functions, which are nonlinear and smooth. Suppose $\dim \mathfrak{g} = N_X, \dim \mathfrak{a} = N_Y$ and noise vectors $\mathbf{n}_k, \mathbf{w}_{k+1}$ are zero-mean Gaussian: $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_k)$ and $\mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{k+1})$.

4.2.0.1 Time Update

- From *a priori* state estimate $\hat{\mathbf{X}}_{k|k}$ and its covariance $\mathbf{P}_{k|k}$, we can extract $2N_X + 1$ sigma points $\mathcal{X}_k^{(i)} \in \mathcal{G}$, $(i = 0, \ldots, 2N_X)$ as follows:

$$\mathcal{X}_k \doteq (\hat{\mathbf{X}}_{k|k}, \quad \exp((\gamma \mathbf{s})^\wedge)\hat{\mathbf{X}}_{k|k}, \quad \exp((-\gamma \mathbf{s})^\wedge)\hat{\mathbf{X}}_{k|k}), \tag{4.2.3}$$

  where γ is a design parameter and $\mathbf{s} \in \mathbb{R}^{N_X}$ is the corresponding column vector of the lower triangular matrix $\mathbf{S}$. Here, $\mathbf{S}$ comes from Cholesky decomposition of $\mathbf{P}_{k|k} = \mathbf{S}\mathbf{S}^T$.

- From Equation (4.2.1), we can define

$$\Upsilon_{k+1}^{(i)} := \mathcal{F}(\mathcal{X}_k^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 2N_X). \tag{4.2.4}$$

Algorithm 4: Weighted Mean $\mathbf{T}$ on matrix Lie group $\mathcal{C}$

Input: a set of matrices $\{\mathbf{Z}_i \in \mathcal{C} | i = 0, \ldots, N_c\}$

1 $\mathbf{T} \leftarrow \mathbf{Z}_0$

2 **for** $k \leftarrow 0$ ***to*** n **do**

3 $\quad \mathbf{\Lambda} \leftarrow \sum_{i=0}^{N_c} w_m^{(i)} \log(\mathbf{Z}_i \mathbf{T}^{-1})$

4 $\quad \mathbf{T} \leftarrow \exp(\mathbf{\Lambda})\, \mathbf{T}$

5 **return** $\mathbf{T}$

By using Algorithm 5, we can compute the weighted mean $\bar{\Upsilon}_{k+1}$ from the propagated sigma points $\Upsilon_{k+1}^{(i)}, (i = 0, \ldots, 2N_X)$. From heuristics, it is enough to set $M = 3$ or 4 in line 2 of Algorithm 5 [16]. The $w_m^{(i)}$ in line 3 represents a set of normalized weights, i.e., $\sum_{i=0}^{2N_X} w_m^{(i)} = 1$. The mean and covariance

of the propagated sigma points are

$$\hat{\mathbf{X}}_{k+1|k} = \bar{\Upsilon}_{k+1} \tag{4.2.5}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{2N_X} w_c^{(i)} \mathbf{q}_i \mathbf{q}_i^T + \mathbf{N}_k, \tag{4.2.6}$$

where $\mathbf{q}_i := (\log(\Upsilon_{k+1}^{(i)} \hat{\mathbf{X}}_{k+1|k}^{-1}))^{\vee} \in \mathbb{R}^{N_X}$.

- A set of sigma points $\mathcal{X}_{k+1}^{(i)}$, $(i = 0, \ldots, 2N_X)$ is redrawn from $\{\Upsilon_{k+1}^{(0)}, \ldots, \Upsilon_{k+1}^{(2N_X)}\}$ by using Equations (4.3.39) and (4.3.40):

$$\mathcal{X}_{k+1} \doteq (\hat{\mathbf{X}}_{k+1|k}, \quad \exp((\gamma \mathbf{u})^{\wedge}) \hat{\mathbf{X}}_{k+1|k}, \quad \exp((-\gamma \mathbf{u})^{\wedge}) \hat{\mathbf{X}}_{k+1|k}), \tag{4.2.7}$$

where $\mathbf{u} \in \mathbb{R}^{N_X}$ is the corresponding column vector of the lower triangular matrix $\mathbf{U}$. Here, $\mathbf{P}_{k+1|k} = \mathbf{U}\mathbf{U}^T$.

4.2.0.2 Measurement Update

- By incorporating $\mathbf{w}_{k+1} = \mathbf{0}$ into Equation (4.2.2), we can define a set of measurement sigma points

$$\mathcal{Y}_{k+1}^{(i)} := \mathcal{H}(\mathcal{X}_{k+1}^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 2N_X). \tag{4.2.8}$$

- By using Algorithm 5, we can compute the mean $\bar{\mathcal{Y}}_{k+1}$ from $\mathcal{Y}_{k+1}^{(i)}$, $(i = 0, \ldots, 2N_X)$. The mean and covariance of the estimated measurement are computed by

$$\hat{\mathbf{Y}}_{k+1} = \bar{\mathcal{Y}}_{k+1} \tag{4.2.9}$$

$$\mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{2N_X} w_c^{(i)} \mathbf{z}_i \mathbf{z}_i^T + \mathbf{W}_{k+1}, \tag{4.2.10}$$

where $\mathbf{z}_i := (\log(\mathcal{Y}_{k+1}^{(i)} \hat{\mathbf{Y}}_{k+1}^{-1}))^{\vee} \in \mathbb{R}^{N_Y}$. The covariance $\mathbf{P_{xy}}$ is given by

$$\mathbf{P_{xy}} = \sum_{i=0}^{2N_X} w_c^{(i)} \mathbf{p}_i \mathbf{z}_i^T, \tag{4.2.11}$$

where $\mathbf{p}_i := (\log(\mathcal{X}_{k+1}^{(i)} \hat{\mathbf{X}}_{k+1|k}^{-1}))^{\vee} \in \mathbb{R}^{N_X}$.

- The Kalman gain is computed by $\mathbf{K} = \mathbf{P_{xy}} \mathbf{P_{yy}}^{-1}$. We can now update the state and covariance according to

$$\hat{\mathbf{X}}_{k+1|k+1} = \exp(\boldsymbol{\phi}^{\wedge}) \hat{\mathbf{X}}_{k+1|k} \tag{4.2.12}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{J}_l(\boldsymbol{\phi})(\mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P_{yy}}\mathbf{K}^T)\mathbf{J}_l(\boldsymbol{\phi})^T, \tag{4.2.13}$$

where $\boldsymbol{\phi} := \mathbf{K}\boldsymbol{\delta} \in \mathbb{R}^{N_X}$ and

$$\boldsymbol{\delta} := (\log(\mathbf{Y}_{k+1} \hat{\mathbf{Y}}_{k+1}^{-1}))^{\vee} \in \mathbb{R}^{N_Y} \tag{4.2.14}$$

Here $\boldsymbol{\delta}$ represents the innovation vector and $\mathbf{J}_l$ denotes the *left* Jacobian of $\mathcal{G}$. The rationale for adopting $\mathbf{J}_l$ in Equation (4.3.50) is given in the next subsection (i.e., Section 4.2.1), from which we find

$$\mathbf{J}_l(\boldsymbol{\phi}) = \int_0^1 \mathbf{C}^{\alpha} d\alpha = \sum_{m=0}^{\infty} \frac{1}{(m+1)!} \left[\mathrm{ad}_{\boldsymbol{\phi}^{\wedge}}\right]^m, \tag{4.2.15}$$

where $\mathbf{C} := \exp(\left[\mathrm{ad}_{\boldsymbol{\phi}^{\wedge}}\right])$ and $\phi \in \mathbb{R}^{N_X}$. In Appendix A.6, we also provide the UKF algorithm, of which state is an element of a matrix Lie group and the measurement is on vector space.

4.2.1 Covariance Update in UKF on a Matrix Lie Group

From Equation (2.3.8) in Section 2.3, the state $\mathbf{X} \in \mathcal{G}$ can be modeled as

$$\mathbf{X} := \exp(\boldsymbol{\varphi}^{\wedge})\, \hat{\mathbf{X}},$$

where $\boldsymbol{\varphi} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\boldsymbol{\varphi}})$ denotes the state error vector and $\hat{\mathbf{X}} \in \mathcal{G}$ is the state estimate. As described in Section 2.3.1, $\mathbf{P}_{\boldsymbol{\varphi}}$ is called the *right* invariant covariance of $\mathbf{X}$.

The state error vector estimated by the time update in Section 4.2.0.1 is assumed to be zero-mean Gaussian, of which covariance, denoted $\mathbf{P}_{k+1|k}$, can be calculated by Equation (4.3.40). However, a special caution is required when computing $\mathbf{P}_{k+1|k+1}$, which is *a posteriori* right invariant covariance of $\mathbf{X}_{k+1}$ after the measurement update. From Section 4.2.0.2, the state $\mathbf{X}_{k+1}$ can be represented as

$$\mathbf{X}_{k+1} = \exp(\boldsymbol{\xi}^{\wedge})\, \hat{\mathbf{X}}_{k+1|k}, \tag{4.2.16}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\phi}, \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T)$ and $\boldsymbol{\phi} = \mathbf{K}\boldsymbol{\delta}$. Note that $\mathbf{P}_{k+1|k+1} \neq \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T$ because generally $\boldsymbol{\phi} \neq \mathbf{0}$. Let us assume that $\mathbf{X}_{k+1}$ can be represented as

$$\mathbf{X}_{k+1} = \exp(\boldsymbol{\epsilon}'^{\wedge})\, \hat{\mathbf{X}}_{k+1|k+1}, \tag{4.2.17}$$

where $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\boldsymbol{\epsilon}'})$. From Equation (4.2.17), we can express $\mathbf{P}_{k+1|k+1} = \mathbf{P}_{\boldsymbol{\epsilon}'}$ because $E(\boldsymbol{\epsilon}') = \mathbf{0}$. The task of this section is now to find $\mathbf{P}_{\boldsymbol{\epsilon}'}$.

Let us define a vector $\boldsymbol{\epsilon} \in \mathbb{R}^{N_X}$ as $\boldsymbol{\epsilon} := \boldsymbol{\xi} - \boldsymbol{\phi}$. Thus, $\boldsymbol{\epsilon}$ has the following distribution: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\boldsymbol{\epsilon}})$, where

$$\mathbf{P}_{\boldsymbol{\epsilon}} = \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T. \tag{4.2.18}$$

Since $\boldsymbol{\xi} = \boldsymbol{\epsilon} + \boldsymbol{\phi}$, Equation (4.2.16) can be rewritten as

$$\mathbf{X}_{k+1} = \exp((\boldsymbol{\epsilon} + \boldsymbol{\phi})^{\wedge})\, \hat{\mathbf{X}}_{k+1|k}. \tag{4.2.19}$$

By substituting Equation (4.3.49) into (4.2.17), we have

$$\mathbf{X}_{k+1} = \exp(\boldsymbol{\epsilon}'^{\wedge}) \exp(\boldsymbol{\phi}^{\wedge})\, \hat{\mathbf{X}}_{k+1|k}. \tag{4.2.20}$$

Combining Equations (4.2.19) and (4.2.20) leads to

$$\exp((\boldsymbol{\epsilon}+\boldsymbol{\phi})^{\wedge}) = \exp(\boldsymbol{\epsilon'}^{\wedge}) \exp(\boldsymbol{\phi}^{\wedge}). \tag{4.2.21}$$

If $\|\boldsymbol{\epsilon'}\| \ll \|\boldsymbol{\phi}\|$, then $\boldsymbol{\epsilon}+\boldsymbol{\phi}$ in Equation (4.2.21) is approximated by

$$\boldsymbol{\epsilon}+\boldsymbol{\phi} \approx \mathbf{J}_l^{-1}(\boldsymbol{\phi})\boldsymbol{\epsilon'}+\boldsymbol{\phi}, \tag{4.2.22}$$

where $\mathbf{J}_l$ denotes the *left* Jacobian of $\mathcal{M}$ [26, 11]. The left Jacobian of $\mathcal{M}$ at $\boldsymbol{\phi}$ is given by

$$\mathbf{J}_l(\boldsymbol{\phi}) = \int_0^1 \mathbf{C}^{\alpha} d\alpha = \sum_{m=0}^{\infty} \frac{1}{(m+1)!} \left[\mathrm{ad}_{\boldsymbol{\phi}^{\wedge}}\right]^m,$$

where $\mathbf{C} := \exp(\left[\mathrm{ad}_{\boldsymbol{\phi}^{\wedge}}\right])$. From Equation (4.2.22), we find

$$\boldsymbol{\epsilon'} \approx \mathbf{J}_l(\boldsymbol{\phi})\boldsymbol{\epsilon},$$

from which we have

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{\boldsymbol{\epsilon'}} \approx \mathbf{J}_l(\boldsymbol{\phi})\mathbf{P}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})\mathbf{J}_l(\boldsymbol{\phi})^T, \tag{4.2.23}$$

where $\mathbf{P}_{\boldsymbol{\epsilon}}$ is given by Equation (4.2.18). This justifies Equation (4.3.50) in Section 4.2.0.2.

Remark 1 If the *left* invariant noise explained in Section 2.3.2 is adopted [10], the *right* Jacobian should be used in the covariance update equation.

4.3 Application: Estimation of Attitudes and Gyro Bias

4.3.1 Direct Product of $SO(3)$ and $\mathbb{R}^3$

Let $\mathcal{M}$ denote a matrix Lie group consisting of matrices of the form

$$\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix},$$

where $\mathbf{R} \in SO(3)$, $\mathbf{B} = \left[\begin{smallmatrix} \mathbf{I}_3 & \mathbf{b} \\ \mathbf{0} & 1 \end{smallmatrix}\right] \in \mathbb{R}^{4\times4}$, and $\mathbf{b} \in \mathbb{R}^3$. Here $\mathbf{0}$ denotes a zero matrix, of which size is clear from the context. Elements of $\mathcal{M}$ will be alternatively represented by an ordered pair $(\mathbf{R}, \mathbf{b})$. In most cases it will be clear from the context which representation is implied. The multiplication and inversion of $\mathcal{M}$ could be understood to be $(\mathbf{R}_1, \mathbf{b}_1) \cdot (\mathbf{R}_2, \mathbf{b}_2) = (\mathbf{R}_1\mathbf{R}_2, \mathbf{b}_1 + \mathbf{b}_2)$ and $(\mathbf{R}, \mathbf{b})^{-1} = (\mathbf{R}^T, -\mathbf{b})$, respectively. Thus, $\mathcal{M}$ can be regarded as the direct product of $SO(3)$ and $\mathbb{R}^3$, of which dimension is six.

The Lie algebra of $\mathcal{M}$, denoted $\mathfrak{m}$, consists of matrices of the form

$$\begin{bmatrix} [\boldsymbol{\omega}] & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi} \end{bmatrix},$$

where $[\boldsymbol{\omega}] \in so(3)$, $\boldsymbol{\Psi} = \left[\begin{smallmatrix} \mathbf{0} & \mathbf{v} \\ \mathbf{0} & 0 \end{smallmatrix}\right] \in \mathbb{R}^{4\times4}$, and $\mathbf{v} \in \mathbb{R}^3$. Element of $\mathfrak{m}$ can also be denoted by $[\boldsymbol{\phi}]$, where $\boldsymbol{\phi} = (\boldsymbol{\omega}, \mathbf{v}) \in \mathbb{R}^6$.

Given $[\boldsymbol{\omega}] \in so(3)$, $\mathbf{R} \in SO(3)$, $\tilde{\boldsymbol{\Psi}} = \left[\begin{smallmatrix} \mathbf{0} & \mathbf{b} \\ \mathbf{0} & 0 \end{smallmatrix}\right]$, $\mathbf{B} = \left[\begin{smallmatrix} \mathbf{I}_3 & \mathbf{b} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$, and $\mathbf{b} \in \mathbb{R}^3$, the exponential mapping from $\mathfrak{m}$ to $\mathcal{M}$ is

$$\exp\left(\begin{bmatrix} [\boldsymbol{\omega}] & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{\Psi}} \end{bmatrix}\right) = \begin{bmatrix} \exp([\boldsymbol{\omega}]) & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix},$$

and the inverse of the exponential map, or logarithm, over the neighborhood near the identity is given by

$$\log\left(\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}\right) = \begin{bmatrix} \log(\mathbf{R}) & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{\Psi}} \end{bmatrix}.$$

An element of a matrix Lie group can be identified with a linear mapping between its Lie algebra via the adjoint representation. For every $\mathbf{X} \in \mathcal{M}$, the adjoint map $\mathrm{Ad}_{\mathbf{X}} : \mathfrak{m} \to \mathfrak{m}$ is defined by $\mathrm{Ad}_{\mathbf{X}}([\boldsymbol{\phi}]) := \mathbf{X}[\boldsymbol{\phi}]\mathbf{X}^{-1}$, where $[\boldsymbol{\phi}] \in \mathfrak{m}$ and $\boldsymbol{\phi} \in \mathbb{R}^6$. If $\mathbf{X} = (\mathbf{R}, \mathbf{b})$ is an element of $\mathcal{M}$, its adjoint map acting on an element

$[\boldsymbol{\phi}] = [(\boldsymbol{\omega}, \mathbf{v})]$ of $\mathfrak{m}$ is given by $\mathrm{Ad}_{\mathbf{X}}([\boldsymbol{\phi}]) = [(\mathbf{R}\boldsymbol{\omega}, \mathbf{v})]$, which also admits the matrix representation $[\mathrm{Ad}_{\mathbf{X}}] \in \mathbb{R}^{6\times 6}$ as

$$[\mathrm{Ad}_{\mathbf{X}}]\,\boldsymbol{\phi} := \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix}. \tag{4.3.24}$$

On matrix Lie algebra $\mathfrak{m}$, the Lie bracket $[\cdot,\cdot] : \mathfrak{m} \times \mathfrak{m} \to \mathfrak{m}$ is given by the matrix commutator: if $\boldsymbol{\Phi}, \boldsymbol{\Gamma} \in \mathfrak{m}$, then $[\boldsymbol{\Phi}, \boldsymbol{\Gamma}] := \boldsymbol{\Phi}\boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}$. Given $\boldsymbol{\Phi} \in \mathfrak{m}$, we can define a linear map $\mathrm{ad}_{\boldsymbol{\Phi}} : \mathfrak{m} \to \mathfrak{m}$ as $\mathrm{ad}_{\boldsymbol{\Phi}}(\boldsymbol{\Gamma}) := [\boldsymbol{\Phi}, \boldsymbol{\Gamma}]$. If $\boldsymbol{\Phi} = [(\boldsymbol{\omega}_1, \mathbf{v}_1)]$ and $\boldsymbol{\Gamma} = [(\boldsymbol{\omega}_2, \mathbf{v}_2)]$ are elements of $\mathfrak{m}$, then $\mathrm{ad}_{\boldsymbol{\Phi}}(\boldsymbol{\Gamma}) = [(\boldsymbol{\omega}_1 \times \boldsymbol{\omega}_2, \mathbf{0})]$, which also admits the matrix representation $[\mathrm{ad}_{\boldsymbol{\Phi}}] \in \mathbb{R}^{6\times 6}$ as

$$[\mathrm{ad}_{\boldsymbol{\Phi}}]\,\boldsymbol{\gamma} := \begin{bmatrix} [\boldsymbol{\omega}_1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_2 \\ \mathbf{v}_2 \end{bmatrix}, \tag{4.3.25}$$

where $[\boldsymbol{\gamma}] = \boldsymbol{\Gamma}$ and $\boldsymbol{\gamma} = (\boldsymbol{\omega}_2, \mathbf{v}_2) \in \mathbb{R}^6$.

4.3.2 Sensor Models and Wahba's Problem

Let $\{\mathcal{B}\}$ and $\{\mathcal{I}\}$ denote the moving body frame fixed to an IMU and the inertial frame fixed to the ground, respectively. The IMU is commonly equipped with gyros, accelerometers, and magnetometers, of which sensing axes are three. Gyros measure the angular velocities expressed in $\{\mathcal{B}\}$. The measurement $\boldsymbol{\omega}^m \in \mathbb{R}^3$ of the gyro can be modeled as

$$\boldsymbol{\omega}^m = \boldsymbol{\omega} + \mathbf{b} + \boldsymbol{\eta}_\omega, \tag{4.3.26}$$

where $\boldsymbol{\omega}$ denotes the ground-truth of an angular rate, $\mathbf{b} \in \mathbb{R}^3$ is the slowly time-varying bias, and $\boldsymbol{\eta}_\omega$ is the zero-mean Gaussian noise.

In quasi-static conditions, accelerometers can measure gravity expressed in $\{\mathcal{B}\}$. If there are no magnetic disturbances nearby the IMU, then magnetometers measure the local geomagnetic field expressed in $\{\mathcal{B}\}$. If the calibrated outputs of the

accelerometer and magnetometer are respectively denoted by $\mathbf{a}, \mathbf{m} \in \mathbb{R}^3$, then we can define $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ as $\mathbf{v}_1 := \mathbf{a}/\|\mathbf{a}\|$ and $\mathbf{v}_2 := \mathbf{m}/\|\mathbf{m}\|$. The directions of gravity and geomagnetic field expressed in $\{\mathcal{I}\}$ are denoted by $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^3$, respectively. We assume that $\mathbf{r}_1, \mathbf{r}_2$ are constant and non-parallel.

The attitude of an IMU can be described by $\mathbf{R} := \mathbf{R}_{\mathcal{IB}} \in SO(3)$, which represents the orientation of $\{\mathcal{B}\}$ with respect to $\{\mathcal{I}\}$. Given a pair of vector measurements $(\mathbf{v}_1, \mathbf{v}_2)$, the optimal attitude $\mathbf{R}^*$ can be determined by solving the following Wahba's problem

$$\mathbf{R}^* = \underset{\mathbf{R} \in SO(3)}{\arg\min} \sum_{i=1}^{2} w_i \|\mathbf{r}_i - \mathbf{R}\mathbf{v}_i\|^2, \tag{4.3.27}$$

where w_i is the positive weight and generally chosen by $w_i = 1/\sigma_i^2$ [50]. As an equivalent form, one can use the normalized weights, $w_i = \sigma_{\text{tot}}^2/\sigma_i^2$, where $1/\sigma_{\text{tot}}^2 := \sum_{i=1}^{2}(1/\sigma_i^2)$ [51]. Here σ_i^2 denotes the variance of $\mathbf{v}_i$ along a direction normal to $E(\mathbf{v}_i)$, where $E(\cdot)$ is the expectation operator. The closed-form solution [52] to Wahba's problem in Equation (4.3.27) is

$$\mathbf{R}^* = \mathbf{V}\mathbf{D}\mathbf{U}^T, \tag{4.3.28}$$

where $\mathbf{U}$ and $\mathbf{V}$ come from the SVD (singular value decomposition) of

$$\mathbf{F} := \sum_{i=1}^{2} w_i \mathbf{v}_i \mathbf{r}_i^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

The matrix $\mathbf{D}$ in Equation (4.3.28) is given by $\mathbf{D} = \text{diag}(1, 1, \det(\mathbf{V}\mathbf{U}^T))$. We remark that there exist numerous alternative solutions [6] to Wahba's problem, which are computationally faster but less robust than Equation (4.3.28).

4.3.3 State Space Equations

In this section, we present the geometric UKF algorithm for estimating attitude and gyro bias. For readers' conveniences, the standard UKF algorithm on vector

space is summarized in Appendix A.5. Consider a dynamical system in discrete time

$$\mathbf{X}_{k+1} = \mathcal{F}(\mathbf{X}_k, \mathbf{n}_k) \in \mathcal{M} \tag{4.3.29}$$

$$\mathbf{Y}_{k+1} = \mathcal{H}(\mathbf{X}_{k+1}, \mathbf{w}_{k+1}) \in SO(3), \tag{4.3.30}$$

where $\mathbf{X}_k \in \mathcal{M}$ is the state variable at time step k, $\mathbf{Y}_{k+1} \in SO(3)$ denotes the measurement matrix at time step $k+1$, $\mathcal{F}$ is the state transition function, and $\mathcal{H}$ represents the measurement function. Suppose zero-mean Gaussian noise vectors $\mathbf{n}_k \in \mathbb{R}^6$ and $\mathbf{w}_{k+1} \in \mathbb{R}^3$ have the following distributions: $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_k)$ and $\mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{k+1})$.

4.3.3.1 State Transition Function

The state $\mathbf{X}_k = (\mathbf{R}_k, \mathbf{b}_k)$ is propagated through $\mathcal{F}$ in Equation (4.3.29) as follows:

$$\mathbf{R}_{k+1} = \exp([\mathbf{n}_k^{(1:3)}])\mathbf{R}_k \exp([\boldsymbol{\omega}_k^m - \mathbf{b}_k]h) \tag{4.3.31}$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \mathbf{n}_k^{(4:6)}, \tag{4.3.32}$$

where h is the integrating time and $\mathbf{n}_k^{(a:b)} \in \mathbb{R}^{b-a+1}$ denotes a vector extracted from the a^{th} to the b^{th} component of $\mathbf{n}_k$. Equation (4.3.31) is constructed by combining the noise model in Equation (2.3.8) and the left invariant dynamics $\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]$, where $\boldsymbol{\omega}$ is described in Equation (4.3.26).

4.3.3.2 Measurement Function

The measurement function $\mathcal{H}$ in Equation (4.3.30) can be designed as

$$\mathbf{Y}_{k+1} = \exp([\mathbf{w}_{k+1}])\mathbf{R}_{k+1}, \tag{4.3.33}$$

where $[\mathbf{w}_{k+1}] \in so(3)$. Here $\mathbf{Y}_{k+1}$ is given by the solution to Wahba's problem as described in Equation (4.3.28). Considering measurement noise covariances, Equation (5.2.1) has several attractive features over existing models [17, 18, 19] as follows:

- There exists a closed-form formula for approximating the covariance of $\mathbf{w}_{k+1}$ (see Lemma 4.2 in Section 4.3.5).
- The angle between gravity and local geomagnetic field is constant. However, many existing models [17, 18] for measurement noise covariances ignore this constraint, while the covariance of $\mathbf{w}_{k+1}$ can consider it.
- Some existing measurement functions [19] assign a unit quaternion to the measurement vector. In this case, the measurement noise covariance may suffer from the rank-deficiency owing to the unit norm constraint. In contrast, the covariance of $\mathbf{w}_{k+1}$ does not have this type of pitfall.

4.3.4 UKF Algorithm for Estimating Attitudes and Gyro Bias

We now derive the intrinsic UKF algorithm for estimating the attitude and gyro bias.

4.3.4.1 Initialization

Let $\hat{\mathbf{X}}_{0|0} = (\hat{\mathbf{R}}_{0|0}, \hat{\mathbf{b}}_{0|0})$ be the initial state estimate. The right invariant covariance of $\hat{\mathbf{X}}_{0|0}$ is denoted by $\hat{\mathbf{P}}_{0|0}$. From Equation (4.3.28), $\hat{\mathbf{R}}_{0|0}$ can be estimated by using initial measurement vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$.

4.3.4.2 Time Update

- From *a priori* state estimate $\hat{\mathbf{X}}_{k|k}$ and its covariance $\mathbf{P}_{k|k}$, we can extract a set of sigma points, denoted $\mathcal{X}_k^{(i)} \in \mathcal{M}$, $(i = 0, \ldots, 12)$ as follows:

$$\mathcal{X}_k^{(0)} = \hat{\mathbf{X}}_{k|k} \tag{4.3.34}$$

$$\mathcal{X}_k^{(i)} = \exp([\gamma \mathbf{s}_i])\, \hat{\mathbf{X}}_{k|k}, \qquad (i = 1, \ldots, 6) \tag{4.3.35}$$

$$\mathcal{X}_k^{(i+6)} = \exp([-\gamma \mathbf{s}_i])\, \hat{\mathbf{X}}_{k|k}, \quad (i = 1, \ldots, 6), \tag{4.3.36}$$

  where $[\gamma \mathbf{s}_i] \in \mathfrak{m}$, $\gamma \in \mathbb{R}$ denotes a design parameter, and $\mathbf{s}_i \in \mathbb{R}^6$ is the i^{th} column vector of the lower triangular matrix $\mathbf{S} \in \mathbb{R}^{6\times 6}$. Here $\mathbf{S}$ is given by Cholesky decomposition of $\mathbf{P}_{k|k} = \mathbf{S}\mathbf{S}^T$. For notational simplicity, Equations (4.3.34), (4.3.35), and (4.3.36) can be collectively rewritten as

$$\mathcal{X}_k \doteq (\hat{\mathbf{X}}_{k|k},\ \exp([\gamma \mathbf{s}])\hat{\mathbf{X}}_{k|k},\ \exp([-\gamma \mathbf{s}])\hat{\mathbf{X}}_{k|k}). \tag{4.3.37}$$

- By setting $\mathbf{n}_k = \mathbf{0}$ in Equation (4.3.29), we can define a set of sigma points, $\mathcal{S}_\Upsilon = \{\Upsilon_{k+1}^{(i)} \in \mathcal{M} | i = 0, \ldots, 12\}$ as

$$\Upsilon_{k+1}^{(i)} := \mathcal{F}(\mathcal{X}_k^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 12), \tag{4.3.38}$$

  where $\mathcal{X}_k^{(i)}$ and $\Upsilon_{k+1}^{(i)}$ can be represented as ordered pairs; $\mathcal{X}_k^{(i)} := (\mathcal{X}_{\mathbf{R},k}^{(i)}, \mathcal{X}_{\mathbf{b},k}^{(i)})$, $\Upsilon_{k+1}^{(i)} := (\Upsilon_{\mathbf{R},k+1}^{(i)}, \Upsilon_{\mathbf{b},k+1}^{(i)}) \in SO(3) \times \mathbb{R}^3$. By substituting Equation (4.3.38) into (4.3.31) and (4.3.32), we have

$$\begin{aligned} \Upsilon_{\mathbf{R},k+1}^{(i)} &= \mathcal{X}_{\mathbf{R},k}^{(i)} \exp([\boldsymbol{\omega}_k^m - \mathcal{X}_{\mathbf{b},k}^{(i)}]h) \\ \Upsilon_{\mathbf{b},k+1}^{(i)} &= \mathcal{X}_{\mathbf{b},k}^{(i)}. \end{aligned}$$

- From Algorithm 5, we can obtain the weighted intrinsic mean

$$\bar{\Upsilon}_{k+1} := (\bar{\Upsilon}_{\mathbf{R},k+1}, \bar{\Upsilon}_{\mathbf{b},k+1})$$

of $\mathcal{S}_\Upsilon$. From heuristics, it is enough to set $n = 3$ or 4 [16] in line 2 of Algorithm 5. The $w_m^{(i)} \in \mathbb{R}$ in line 3 represents a set of normalized weights, which satisfies $\sum_{i=0}^{12} w_m^{(i)} = 1$.

Algorithm 5: Weighted Mean $\mathbf{T}$ on matrix Lie group $\mathcal{M}$

Input: a set of matrices $\{\mathbf{Z}_i \in \mathcal{M} | i = 0, \ldots, 12\}$

1 $\mathbf{T} \leftarrow \mathbf{Z}_0$

2 **for** $k \leftarrow 0$ ***to*** n **do**

3 $\quad \mathbf{\Lambda} \leftarrow \sum_{i=0}^{12} w_m^{(i)} \log(\mathbf{Z}_i \mathbf{T}^{-1})$

4 $\quad \mathbf{T} \leftarrow \exp(\mathbf{\Lambda})\, \mathbf{T}$

5 **return** $\mathbf{T}$

- The mean and covariance of the propagated sigma points $\Upsilon_{k+1}^{(i)}, i = 0, \ldots, 12$ are

$$\hat{\mathbf{X}}_{k+1|k} = \bar{\Upsilon}_{k+1} \tag{4.3.39}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{12} w_c^{(i)} \mathbf{q}_i \mathbf{q}_i^T + \mathbf{N}_k, \tag{4.3.40}$$

where $[\mathbf{q}_i] := \log(\Upsilon_{k+1}^{(i)}\, \hat{\mathbf{X}}_{k+1|k}^{-1}) \in \mathfrak{m}$ and $w_c^{(i)} \in \mathbb{R}$ is a weighting factor. The process noise covariance in Equation (4.3.40) can be set to the form as $\mathbf{N}_k = \begin{bmatrix} c\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & d\mathbf{I}_3 \end{bmatrix}$, where $c, d \in \mathbb{R}$ represent tuning parameters [17, 53]. For further explanations, $\hat{\mathbf{X}}_{k+1|k}$ in Equation (4.3.39) can be written as

$$\hat{\mathbf{X}}_{k+1|k} := (\hat{\mathbf{R}}_{k+1|k}, \hat{\mathbf{b}}_{k+1|k}), \tag{4.3.41}$$

where $\hat{\mathbf{R}}_{k+1|k} \in SO(3)$ and $\hat{\mathbf{b}}_{k+1|k} \in \mathbb{R}^3$.

- A set of sigma points $\mathcal{X}_{k+1}^{(i)}$, $(i = 0, \ldots, 12)$ is redrawn from $\Upsilon_{k+1}^{(i)}$ by using

Equations (4.3.39) and (4.3.40):

$$\mathcal{X}_{k+1} \doteq (\hat{\mathbf{X}}_{k+1|k}, \exp([\gamma\mathbf{u}])\hat{\mathbf{X}}_{k+1|k}, \exp([-\gamma\mathbf{u}])\hat{\mathbf{X}}_{k+1|k}), \tag{4.3.42}$$

where $\mathbf{u} \in \mathbb{R}^6$ is the corresponding column vector of the lower triangular matrix $\mathbf{U}$. Here, $\mathbf{P}_{k+1|k} = \mathbf{U}\mathbf{U}^T$.

4.3.4.3 Measurement Update

- When the IMU moves with high acceleration or near the magnetic disturbing environments, the measurements of accelerometers and magnetometers can be corrupted by disturbances. In Appendix A.7, the existing methods for handling these disturbances are provided.

- By setting $\mathbf{w}_{k+1} = \mathbf{0}$ in Equation (4.3.30), we can define a set of measurement sigma points, $\mathcal{S}_\mathcal{Y} = \{\mathcal{Y}_{k+1}^{(i)} \in SO(3) | i = 0, \ldots, 12\}$ as

$$\mathcal{Y}_{k+1}^{(i)} := \mathcal{H}(\mathcal{X}_{k+1}^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 12), \tag{4.3.43}$$

where $\mathcal{X}_{k+1}^{(i)} := (\mathcal{X}_{\mathbf{R},k+1}^{(i)}, \mathcal{X}_{\mathbf{b},k+1}^{(i)}) \in SO(3) \times \mathbb{R}^3$ is given by Equation (4.3.42). By combining Equations (5.2.1) and (4.3.43), we have

$$\mathcal{Y}_{k+1}^{(i)} = \mathcal{X}_{\mathbf{R},k+1}^{(i)} \quad (i = 0, \ldots, 12). \tag{4.3.44}$$

- The mean of $\mathcal{Y}_{k+1}^{(0)}, \ldots, \mathcal{Y}_{k+1}^{(12)}$, denoted $\hat{\mathbf{Y}}_{k+1}$, is simply

$$\hat{\mathbf{Y}}_{k+1} = \hat{\mathbf{R}}_{k+1|k}, \tag{4.3.45}$$

where $\hat{\mathbf{R}}_{k+1|k}$ is given by Equations (4.3.39) and (4.3.41). The covariance of $\mathcal{Y}_{k+1}^{(0)}, \ldots, \mathcal{Y}_{k+1}^{(12)}$ is

$$\mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{12} w_c^{(i)} \mathbf{z}_i \mathbf{z}_i^T + \mathbf{W}_{k+1}, \tag{4.3.46}$$

where $[\mathbf{z}_i] := \log(\mathcal{Y}_{k+1}^{(i)} \hat{\mathbf{Y}}_{k+1}^{-1}) \in so(3)$ and $\mathbf{W}_{k+1}$ is the right invariant covariance of the solution to Wahba's problem. In the next section, we will show that $\mathbf{W}_{k+1}$ can be obtained offline by using Equation (4.3.56). The covariance $\mathbf{P_{xy}}$ is calculated by

$$\mathbf{P_{xy}} = \sum_{i=0}^{12} w_c^{(i)} \mathbf{p}_i \mathbf{z}_i^T, \tag{4.3.47}$$

where $[\mathbf{p}_i] := \log(\mathcal{X}_{k+1}^{(i)} \hat{\mathbf{X}}_{k+1|k}^{-1}) \in \mathfrak{m}$.

- The Kalman gain is computed by $\mathbf{K} = \mathbf{P_{xy}} \mathbf{P_{yy}}^{-1}$. Let us define the innovation vector $\boldsymbol{\delta} \in \mathbb{R}^3$ as

$$[\boldsymbol{\delta}] := \log(\mathbf{Y}_{k+1} \hat{\mathbf{Y}}_{k+1}^{-1}) \in so(3). \tag{4.3.48}$$

Recall that $\mathbf{Y}_{k+1}$ and $\hat{\mathbf{Y}}_{k+1}$ are given by Equations (4.3.28) and (4.3.45), respectively. We can now update the state and covariance according to

$$\hat{\mathbf{X}}_{k+1|k+1} = \exp([\boldsymbol{\phi}])\, \hat{\mathbf{X}}_{k+1|k} \tag{4.3.49}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{J}_l(\boldsymbol{\phi})(\mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P_{yy}}\mathbf{K}^T)\mathbf{J}_l(\boldsymbol{\phi})^T, \tag{4.3.50}$$

where $\boldsymbol{\phi} := \mathbf{K}\boldsymbol{\delta} \in \mathbb{R}^6$, $[\boldsymbol{\phi}] \in \mathfrak{m}$, and $\mathbf{J}_l$ denotes the *left* Jacobian of $\mathcal{M}$. The rationale for adopting $\mathbf{J}_l$ in Equation (4.3.50) is provided in Section 4.2.1. The left Jacobian of $\mathcal{M}$ at $\boldsymbol{\phi}$ is given by

$$\mathbf{J}_l(\boldsymbol{\phi}) = \int_0^1 \mathbf{C}^\alpha d\alpha = \sum_{m=0}^{\infty} \frac{1}{(m+1)!} \left[\mathrm{ad}_{[\boldsymbol{\phi}]}\right]^m, \tag{4.3.51}$$

where $\mathbf{C} := \exp(\left[\mathrm{ad}_{[\boldsymbol{\phi}]}\right])$. From Equation (4.3.25), we have

$$\left[\mathrm{ad}_{[\boldsymbol{\phi}]}\right] = \begin{bmatrix} [\boldsymbol{\phi}^{(1:3)}] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{4.3.52}$$

where $[\boldsymbol{\phi}^{(1:3)}] \in so(3)$. Here $\boldsymbol{\phi}^{(1:3)} \in \mathbb{R}^3$ is a vector extracted from the first to the third component of $\boldsymbol{\phi}$. The analytic expression for $\mathbf{J}_l(\boldsymbol{\phi})$ in Equation (4.3.51) is available as

$$\mathbf{J}_l(\boldsymbol{\phi}) = \begin{bmatrix} \mathbf{Q}(\boldsymbol{\phi}^{(1:3)}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}, \tag{4.3.53}$$

where $\mathbf{Q}(\boldsymbol{\phi}^{(1:3)})$ denotes the left Jacobian of $SO(3)$ at $\boldsymbol{\phi}^{(1:3)}$. Here $\mathbf{Q}(\boldsymbol{\phi}^{(1:3)})$ is given by

$$\mathbf{Q}(\mathbf{a}) = \mathbf{I}_3 + \left(\frac{1 - \cos\|\mathbf{a}\|}{\|\mathbf{a}\|^2}\right)[\mathbf{a}] + \left(\frac{\|\mathbf{a}\| - \sin\|\mathbf{a}\|}{\|\mathbf{a}\|^3}\right)[\mathbf{a}]^2,$$

where $\mathbf{a} := \boldsymbol{\phi}^{(1:3)}$.

4.3.5 Measurement Noise Covariance

This section presents the algorithm for determining the measurement noise covariance from actual measurements of gravity and geomagnetic field by using accelerometers and magnetometers.

4.3.5.1 Covariances of the Solution to Wahba's Problem

In [50], Shuster provides a useful formula for the left invariant covariance of the solution to Wahba's problem in Equation (4.3.27) as follows:

Proposition 4.1. *The left invariant covariance of* $\mathbf{R}$ *in Equation (4.3.27) can be approximated to the first order as*

$$\left(\sum_{i=1}^{2} \frac{1}{\sigma_i^2}(\mathbf{I}_3 - \bar{\mathbf{A}}\mathbf{r}_i\mathbf{r}_i^T\bar{\mathbf{A}}^T)\right)^{-1}, \tag{4.3.54}$$

where $\bar{\mathbf{A}} \in SO(3)$ *denotes the true value of* $\mathbf{R}^T$*, which is usually unknown.* $\bar{\mathbf{A}}$ *can be approximated by*

$$\bar{\mathbf{A}} \approx \underset{\mathbf{A}\in SO(3)}{\arg\min} \sum_{i=1}^{2} \frac{1}{\sigma_i^2} \|\mathbf{v}_i - \mathbf{A}\mathbf{r}_i\|^2. \tag{4.3.55}$$

Proof. In [50], it is said that the left invariant covariance of $\mathbf{R}$ is indeed the inverse of the Fisher information matrix without sufficient explanations. In Appendix A.9, we provide an in-depth proof by computing the Cramer-Rao lower bound (CRLB). □

Lemma 4.2. *From Equation (4.3.54), the right invariant covariance of* $\mathbf{R}$ *is given by*

$$(\sum_{i=1}^{2} \frac{1}{\sigma_i^2}(\mathbf{I}_3 - \mathbf{r}_i\mathbf{r}_i^T))^{-1} \tag{4.3.56}$$

Proof. A straightforward calculation using the relation given in Equation (2.3.10) yields the result. □

Note that the left invariant covariance of $\mathbf{R}$ in Equation (4.3.54) is equivalent to the covariance of the solution to Wahba's problem represented with respect to the "IMU body frame". In contrast, the right invariant covariance of $\mathbf{R}$ in Equation (4.3.56) is the covariance of the solution to Wahba's problem represented with respect to the "ground-fixed frame".

If parameters $\sigma_i^2, \mathbf{r}_i$ are given, the right invariant covariance of $\mathbf{R}$ in Equation (4.3.56) can be uniquely determined as a constant matrix, which is independent from $\bar{\mathbf{A}}$. However, the left invariant covariance of $\mathbf{R}$ in Equation (4.3.54) requires $\bar{\mathbf{A}}$ as well as σ_i^2 and $\mathbf{r}_i$.

Let us assume that the IMU is moving. Obviously, $\bar{\mathbf{A}}$ is also changing. In this situation, the left invariant covariance of $\mathbf{R}$ should be updated at every time step

by computing the inverse of matrix according to varying $\bar{\mathbf{A}}$ while the right invariant covariance is invariant. Furthermore, when the IMU is rotating, $\bar{\mathbf{A}}$ is highly inaccurate because $\mathbf{v}_1$ and $\mathbf{v}_2$ are inevitably noisy.

Remark 2 For fast and accurate computation of the measurement noise covariance in UKF, the proposed formula given by Equation (4.3.56) is more suitable than Shuster's result in (4.3.54).

4.3.5.2 Determination of Parameters in the Covariance of R

In this section, we present how to determine parameters in Equation (4.3.56) such as σ_i^2 and $\mathbf{r}_i$ $(i = 1, 2)$ offline.

- **Constant Vectors** $(\mathbf{r}_1, \mathbf{r}_2)$**:** Without loss of generality, we can assign each axis of the inertial frame $\{\mathcal{I}\}$ as follows: The negative direction of gravity is set to be the y-axis of $\{\mathcal{I}\}$. The x-axis of $\{\mathcal{I}\}$ is set to be orthogonal to both gravity and geomagnetic field. In this setting, $\mathbf{r}_1 = (0, 1, 0)^T$ and

$$\mathbf{r}_2 = (0, \cos(\phi), \sin(\phi))^T, \tag{4.3.57}$$

  where ϕ is unknown and should be determined.

  We assume that the IMU is stationary and multiple measurement pairs are given. From Proposition A.2 in Appendix A.8, $\hat{\mathbf{v}}_i := E(\mathbf{v}_i)$ can be calculated for $i = 1, 2$. Since $\mathbf{r}_1^T \mathbf{r}_2 \approx \hat{\mathbf{v}}_1^T \hat{\mathbf{v}}_2$, we can approximate

$$\phi \approx \cos^{-1}(\hat{\mathbf{v}}_1^T \hat{\mathbf{v}}_2). \tag{4.3.58}$$

- **Variances** (σ_1^2, σ_2^2)**:** Let $\breve{\mathbf{v}}_i$ denote the truth of noisy unit vector $\mathbf{v}_i, (i = 1, 2)$, where $\|\breve{\mathbf{v}}_i\| = 1$. The covariance of $\mathbf{v}_i$ is

$$\mathbf{M}_t = \sigma_i^2 (\mathbf{I}_3 - \breve{\mathbf{v}}_i \breve{\mathbf{v}}_i^T). \tag{4.3.59}$$

Let the SVD of $\mathbf{M}_t$ be given by $\mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^T$, Theoretically, $\mathbf{\Sigma}_t = \text{diag}(\sigma_i^2, \sigma_i^2, 0)$ and $\breve{\mathbf{v}}_i$ is in the direction of singular value 0.

However, in a real situation, the ground-truth of $\breve{\mathbf{v}}_i$ is unavailable and σ_i^2 should be determined differently. We assume that the IMU is stationary and N measurements are given. The covariance of $\mathbf{v}_i$ can be estimated by

$$\mathbf{M}_a = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{v}_i^{(j)} - \hat{\mathbf{v}}_i)(\mathbf{v}_i^{(j)} - \hat{\mathbf{v}}_i)^T, \tag{4.3.60}$$

where $\mathbf{v}_i^{(j)}$ denotes the j^{th} measurement vector coming from the i^{th} sensor: (the first sensor: accelerometer; the second sensor: magnetometer). Let the SVD of $\mathbf{M}_a$ be given by $\mathbf{U}_a\mathbf{\Sigma}_a\mathbf{V}_a^T$, where $\mathbf{\Sigma}_a = \text{diag}(s_1, s_2, s_3)$ and $s_1 \geq s_2 \geq s_3 \approx 0$. We expect $\mathbf{\Sigma}_a$ will be close to its theoretical value $\mathbf{\Sigma}_t$. From this, we can set

$$\sigma_i^2 = \frac{tr(\mathbf{M}_a)}{2}. \tag{4.3.61}$$

4.3.6 Experimental Results

In this subsection, we compare the performance of the proposed algorithm ("UKF on $SO(3)$") against the state-of-the-art methods including the quaternion-based UKF ("UKF on Quaternion") [18], the quaternion-based EKF ("EKF on Quaternion") [54], and the passive complementary filter in nonlinear complementary filters (NCF) on $SO(3)$ ("NCF on $SO(3)$") [20] in terms of both convergence rate and estimation accuracy of attitudes and gyro biases by using synthetic and real data.

In simulation (or real experiment), we first generate (or collect) two sets of

noisy sensor data in conditions of the stationary and moving IMU. For fair comparison, we tune the parameters of each filter such as noise covariances or feedback gains so that the filters have the similar performance of noise filtering in a "stationary" condition. After tuning parameters of each filter, the other dataset assuming the condition of the "moving" IMU can be used to evaluate the convergence rate and the accuracy of the filters.

For notational simplicity and consistency, the above two sets of synthetic (or real) data will be identically denoted by $\mathcal{S} := \{(\boldsymbol{\omega}_k^m, \mathbf{v}_{1,k}, \mathbf{v}_{2,k}) | k = 1, \ldots, N\}$. From the context, it is clear which set the above $\mathcal{S}$ actually corresponds to. The ground-truths of the attitude and gyro bias are denoted $\breve{\mathbf{R}}_k$ and $\breve{\mathbf{b}}$, respectively. In simulation and real experiments, the time intervals of sampling and filter updates are set equal to $h_0 = 1/60$ sec. The weighting factors, $w_m^{(i)}$ and $w_c^{(i)}, i = 0, \ldots, 12$ in our UKF are given by Equations (A.5.13) and (A.5.14).

4.3.6.1 Simulation

In this simulation, the vectors in Equation (4.3.27) are set to be $\mathbf{r}_1 = (0, 1, 0)^T$ and $\mathbf{r}_2 = (0, \cos(\phi_s), \sin(\phi_s))^T$, where $\phi_s = 2.4$ radian. We can randomly set the ground-truth, $\breve{\mathbf{R}}_1 \in SO(3)$ of an initial attitude.

- **Stationary IMU:** Let us assume the IMU is "stationary". In this condition, the ground-truth of an angular rate is obviously a zero vector: $\breve{\boldsymbol{\omega}}_k = \mathbf{0}, k = 1, \ldots, N$, where $N = 2640$. By assuming the ground-truth of a gyro bias is

Table 4.1: Constants for specifying noise level in simulation

	$\mu_{\mathbf{R}}$	$\gamma_{\mathbf{R}}$	$\mu_{\mathbf{b}}$ (rad/sec.)	$\gamma_{\mathbf{b}}$ (rad/sec.)
simulation	0.6658°	0.0008°	0.0178	0.0001

a zero vector, a set of synthetic data is generated by

$$\boldsymbol{\omega}_k^m = \boldsymbol{\eta}_{\omega,k} \tag{4.3.62}$$

$$\mathbf{v}_{1,k} = (\breve{\mathbf{R}}_1^T \mathbf{r}_1 + \boldsymbol{\eta}_{\mathbf{v}_{1,k}})/\|\breve{\mathbf{R}}_1^T \mathbf{r}_1 + \boldsymbol{\eta}_{\mathbf{v}_{1,k}}\| \tag{4.3.63}$$

$$\mathbf{v}_{2,k} = (\breve{\mathbf{R}}_1^T \mathbf{r}_2 + \boldsymbol{\eta}_{\mathbf{v}_{2,k}})/\|\breve{\mathbf{R}}_1^T \mathbf{r}_2 + \boldsymbol{\eta}_{\mathbf{v}_{2,k}}\|, \tag{4.3.64}$$

where $\boldsymbol{\eta}_{\omega,k} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I}_3), \boldsymbol{\eta}_{\mathbf{v}_{1,k}} \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_3)$, and $\boldsymbol{\eta}_{\mathbf{v}_{2,k}} \sim \mathcal{N}(\mathbf{0}, \sigma_2^2 \mathbf{I}_3)$ represent the Gaussian noise vectors. Here $\sigma_0 = (1/h_0)0.002$ radian/sec., $\sigma_1 = 0.01$, and $\sigma_2 = 0.0158$.

If the IMU is motionless, the noises of the attitude and gyro bias can be defined as

$$\epsilon_{\mathbf{R},k} := (180^\circ/\pi)\|\log(\breve{\mathbf{R}}_1^{-1}\hat{\mathbf{R}}_{k|k})\|$$
$$\epsilon_{\mathbf{b},k} := \|\hat{\mathbf{b}}_{k|k} - \breve{\mathbf{b}}_k\|,$$

where $\epsilon_{\mathbf{R},k}$ and $\epsilon_{\mathbf{b},k}$ represent the noises of the estimated attitude and gyro bias at time step k, respectively.

By using the synthetic data given in Equations (4.3.62)-(4.3.64), we can tune the parameters i.e. the noise covariances or gains of each filter so as to simultaneously satisfy

$$|\bar{\epsilon}_{\mathbf{R}} - \mu_{\mathbf{R}}| < \gamma_{\mathbf{R}} \tag{4.3.65}$$

$$|\bar{\epsilon}_{\mathbf{b}} - \mu_{\mathbf{b}}| < \gamma_{\mathbf{b}}, \tag{4.3.66}$$

where $\bar{\epsilon}_{\mathbf{R}} := \frac{1}{N}\sum_{k=1}^{N}\epsilon_{\mathbf{R},k}$ and $\bar{\epsilon}_{\mathbf{b}} := \frac{1}{N}\sum_{k=1}^{N}\epsilon_{\mathbf{b},k}$. Table 4.1 shows the specific values assigned to constants in Equation (4.3.65) and (4.3.66) for the simulation.

- **Moving IMU:** We assume that the IMU is "moving". In this condition, the convergence rate and accuracy of each filter is evaluated. For realistic simulation, we first collect a set of real angular rate data denoted $\breve{\boldsymbol{\omega}}_k$ from the actual gyro (L3G4200D) with a sampling time period h_0.

  From $\breve{\mathbf{R}}_1$, the true attitudes can be iteratively generated by

  $$\breve{\mathbf{R}}_{k+1} = \breve{\mathbf{R}}_k \exp([\breve{\boldsymbol{\omega}}_k]h_0).$$

  We now generate synthetic data as follows:

  $$\begin{aligned}
  \boldsymbol{\omega}_k^m &= \breve{\boldsymbol{\omega}}_k + \breve{\mathbf{b}} + \boldsymbol{\eta}_{\omega,k} \\
  \mathbf{v}_{1,k} &= (\breve{\mathbf{R}}_k^T\mathbf{r}_1 + \boldsymbol{\eta}_{\mathbf{v}_{1,k}})/\|\breve{\mathbf{R}}_k^T\mathbf{r}_1 + \boldsymbol{\eta}_{\mathbf{v}_{1,k}}\| \\
  \mathbf{v}_{2,k} &= (\breve{\mathbf{R}}_k^T\mathbf{r}_2 + \boldsymbol{\eta}_{\mathbf{v}_{2,k}})/\|\breve{\mathbf{R}}_k^T\mathbf{r}_2 + \boldsymbol{\eta}_{\mathbf{v}_{2,k}}\|,
  \end{aligned}$$

  where $k = 1, \ldots, N$. The ground-truth of the gyro bias is set to be $\breve{\mathbf{b}} = (-0.06, 0.3, 0.3)^T$ radian/sec.

  To simulate the large initial estimation errors of gyro bias and attitude, we set $\hat{\mathbf{b}}_{1|1} = \mathbf{0}$ and $\hat{\mathbf{R}}_{1|1} = \breve{\mathbf{R}}_1 \exp([\mathbf{a}_1])$, where $\mathbf{a}_1 = (3.13/\sqrt{3})(1,1,1)^T$. Given tuning parameters, we can run each filter. Let us define

  $$s_k := (180^\circ/\pi)\|\log \breve{\mathbf{R}}_k^{-1}\hat{\mathbf{R}}_{k|k}\| \tag{4.3.67}$$

  $$d_k := \|\hat{\mathbf{b}}_{k|k} - \breve{\mathbf{b}}\|, \tag{4.3.68}$$

  where s_k and d_k represent the estimation errors of the attitude and gyro bias at time step k, respectively.

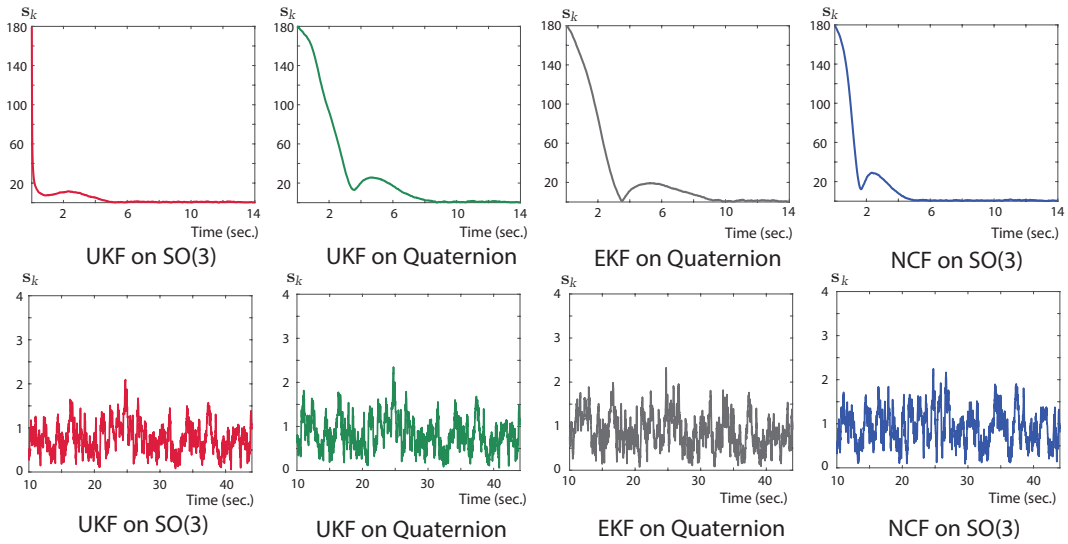


Figure 4.1: Simulation: errors of attitude estimates (in degrees) during time $t \in [0, 14]$ sec. (top); and during $t \in [10, 44]$ sec. (bottom).

Table 4.2: Results of simulations: time $t \in [10, 44]$ sec.

Average of attitude errors (in degrees)				Average of gyro bias errors (in radian/sec.)			
UKF on $SO(3)$	Quaternion UKF	Quaternion EKF	NCF on $SO(3)$	**UKF on** $SO(3)$	Quaternion UKF	Quaternion EKF	NCF on $SO(3)$
0.7594	0.8288	0.8961	0.9284	**0.0179**	0.0183	0.0186	0.0188

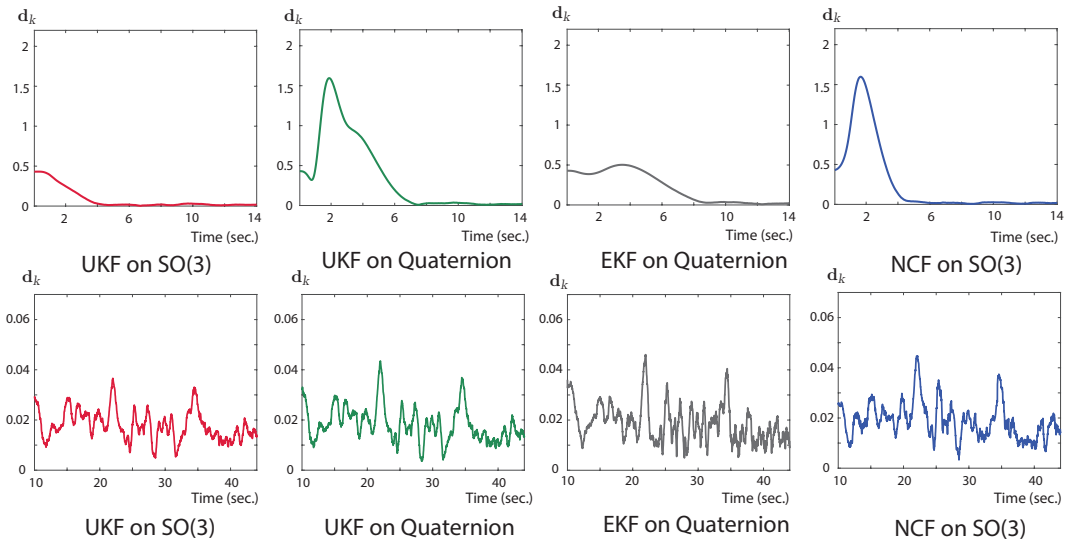


Figure 4.2: Simulation: errors of gyro bias estimates (in radian/sec.) during time $t \in [0, 14]$ sec. (top); and during $t \in [10, 44]$ sec. (bottom).

From Figures 4.1 and 4.2, one can find that the performance of proposed method ("UKF on $SO(3)$") is superior to existing state-of-the-art methods in terms of both convergence rate and estimation accuracy. Moreover, the conventional filters show unfavorable salient overshoots. Table 4.2 summarizes the results of the simulation.

4.3.6.2 Real Experiments

The IMU for real experiments consists of L3G4200D (gyro), LIS3LV02DQ (accelerometer), HMC5883L (magnetometer) and Cortex-M3$^{\text{TM}}$ (microcontroller). To obtain the ground-truths of attitudes, $\breve{\mathbf{R}}_k$ at time step k, we use an optical motion capture system, OptiTrack$^{\text{TM}}$ consisting of multiple networked infrared cameras. For fair comparison among filters, we perform experiments with real data in a

Table 4.3: Constants for specifying noise level in real experiments

	$\mu_{\mathbf{R}}$	$\gamma_{\mathbf{R}}$	$\mu_{\mathbf{b}}$ (rad/sec.)	$\gamma_{\mathbf{b}}$ (rad/sec.)
real experiment	0.2926°	0.0002°	0.0014	0.0001

condition of negligible disturbances.

- **Stationary IMU:** In a similar way to simulation, we first determine the tuning parameters of each filter according to the performance of noise filtering in a "stationary" condition. By putting the IMU on the palm of one's hand in a stationary condition, we can collect a set of IMU sensor data with a sampling time period h_0.

  If the IMU is rotating, the slowly time-varying gyro bias is initially unknown. However, if the IMU is in a static condition, the gyro bias can be temporarily captured by averaging a set of gyro data within a certain time interval [55, 56]. In this experiment, the captured gyro bias in a static condition is assumed to be the ground-truth of the gyro bias denoted $\breve{\mathbf{b}}$ for a short time duration.

  By using Equation (4.3.61), we can compute the variance, σ_i^2 of a unit vector, $\mathbf{v}_{i,k}$, $i = 1, 2$. In this experiment, we obtain $\sigma_1^2 = 0.895 \times 10^{-4}$ and $\sigma_2^2 = 1.911 \times 10^{-4}$. Let us denote by ϕ_r the angle between $\mathbf{r}_1$ and $\mathbf{r}_2$: $\phi_r = \cos^{-1}(\mathbf{r}_1^T \mathbf{r}_2)$, In our experiment, we can find $\phi_r = 2.486$ radian by using Proposition A.2 in Appendix A.8 and Equation (4.3.58).

  After setting constants in Equations (4.3.65) and (4.3.66) as given in Table 4.3, we can determine the tuning parameters of each filter.

- **Moving IMU:** We first attach the IMU and four reflective markers to a plastic plate rigidly. A set of real data $\{(\boldsymbol{\omega}_k^m, \mathbf{v}_{1,k}, \mathbf{v}_{2,k}) | k = 1, \ldots, N_r\}$ coming from the IMU and the ground-truth attitude $\breve{\mathbf{R}}_k$ obtained by OptiTrack$^{\text{TM}}$ infrared camera system are saved into files synchronously. Here the number of measurements $N_r = 3000$.

  To evaluate the convergence rate and accuracy of each filter when the initial estimation errors of the gyro bias and attitude are large, we set the initial estimates as follows: $\hat{\mathbf{b}}_{1|1} = \breve{\mathbf{b}} + (1/h_0)(-0.001, 0.005, 0.005)^T = \breve{\mathbf{b}} + (-0.06, 0.3, 0.3)^T$ (radian/sec.) and $\hat{\mathbf{R}}_{1|1} \leftarrow \breve{\mathbf{R}}_1 \exp([\mathbf{a}_1])$, where $\mathbf{a}_1 = (3.13/\sqrt{3})(1, 1, 1)^T$. Recall that $\breve{\mathbf{b}}$ can be obtained in the previous condition of the stationary IMU.

  Figures 4.3 and 4.4 show the errors, $\mathbf{s}_k$ and $\mathbf{d}_k$, $k = 1 \ldots, N_r$, which are defined in Equations (4.3.67) and (4.3.68). Table 4.4 summarizes the results of real experiments. As shown in Figures 4.3, 4.4 and Table 4.4, the proposed method ("UKF on $SO(3)$") demonstrates the superior performance to existing methods in terms of estimation accuracy. Like the result of simulation, "UKF on $SO(3)$" is the most rapidly converging whereas the other methods show slow convergence rate and some salient overshoots.

Table 4.4: Results of real experiments: time $t \in [10, 50]$ sec.

Average of attitude errors (in degrees)				Average of gyro bias errors (in radian/sec.)			
UKF on $SO(3)$	Quaternion UKF	Quaternion EKF	NCF on $SO(3)$	**UKF on** $SO(3)$	Quaternion UKF	Quaternion EKF	NCF on $SO(3)$
2.3587	2.6663	2.6947	2.7972	**0.0106**	0.0121	0.0125	0.0128

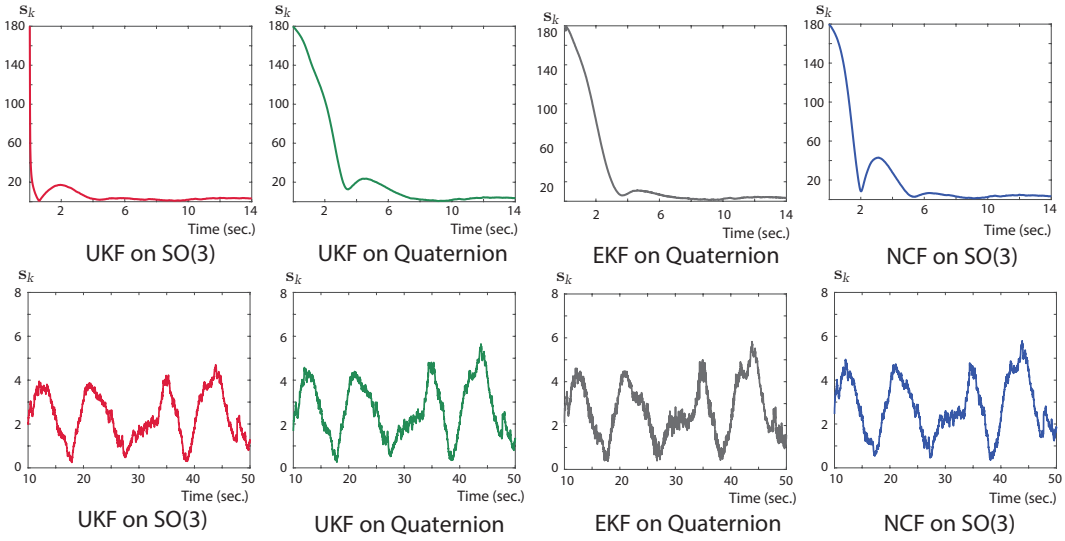


Figure 4.3: Real experiments: errors of attitude estimates (in degrees) during time $t \in [0, 14]$ sec (top); and during $t \in [10, 50]$ sec. (bottom).

Table 4.5: Average computation time of each filter (in μsec.)

	UKF on $SO(3)$	UKF on Quaternion	EKF on Quaternion	NCF on $SO(3)$
Average time	8.1	7.9	6.8	0.2

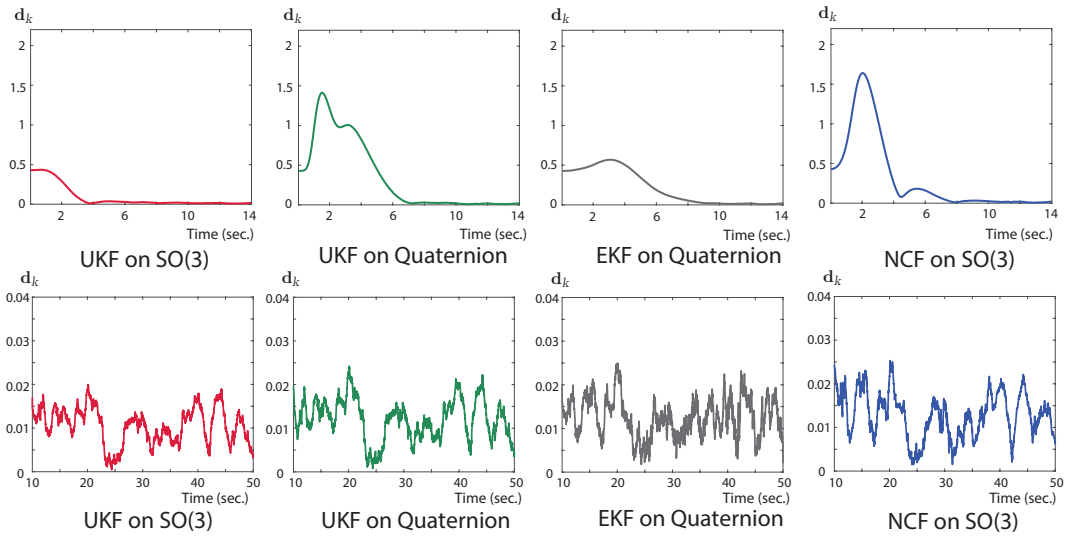


Figure 4.4: Real experiments: errors of gyro bias estimates (in radian/sec.) during time $t \in [0, 14]$ sec (top); and during $t \in [10, 50]$ sec. (bottom).

5
Conclusion

In this thesis, we have presented two different geometric algorithms for fusing sensors: two-frame sensor calibration and state estimation.

5.1 Two-Frame Sensor Calibration

This thesis has presented a fast and numerically robust local optimization algorithm for the two-frame sensor calibration problem. Using coordinate-invariant differential geometric methods that take into account the matrix Lie group structure of the rigid-body transformations, the proposed local descent method makes use of analytic gradients and Hessians, and a strictly descending fast step-size estimate to achieve significant performance improvements. As a second contribution, a two-phase stochastic geometric optimization algorithm for finding a stochastic global minimizer is derived based on our earlier local optimizer. In both cases the calibration problem is formulated as a minimization of the objective function

$\sum_i \|\mathbf{A}_i\mathbf{X} - \mathbf{Y}\mathbf{B}_i\|^2$ on the space $SE(3) \times SE(3)$. After deriving necessary and sufficient conditions for the existence and uniqueness of exact solutions, we perform both synthetic and real experiments that verify the advantages of our stochastic global optimization method over existing local quaternion-based methods.

In this thesis, we also present several applications of the proposed two-frame sensor calibration algorithm such as camera-marker calibration for unmanned aerial vehicle, head-eye calibration for humanoid robot, and registration for improving the accuracy of eye trackers.

5.2 Unscented Kalman Filtering for Estimation of Attitude and Gyro Bias

An geometric (intrinsic) UKF algorithm has been presented to estimate the attitude and gyro bias. In our formulation, the measurement is given by the solution to Wahba's problem, of which form is $SO(3)$ as follows.

$$\mathbf{Y}_{k+1} = \exp([\mathbf{w}_{k+1}])\mathbf{R}_{k+1}, \tag{5.2.1}$$

where $[\mathbf{w}_{k+1}] \in so(3)$ and $\mathbf{Y}_{k+1}$ is given by the solution to Wahba's problem.

Furthermore, an offline algorithm for computing the covariance of the solution to Wahba's problem that is represented with respect to the ground-fixed frame has been explained. Through simulation and real experiments, the proposed algorithm has been shown to be superior to existing estimators in terms of convergence behavior and estimation accuracy.

A

Appendix

A.1 Existence and Uniquess of Solutions to $\mathbf{AX} = \mathbf{YB}$ on $SE(3)$

A.1.1 Proof of Proposition 3.1

Proof. First, note that the general form of the solution to $\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_B$ is given by $\mathbf{R}_X = \mathbf{R}_A^T\boldsymbol{\Theta}$, $\mathbf{R}_Y = \boldsymbol{\Theta}\mathbf{R}_B^T$, where $\boldsymbol{\Theta} = \mathbf{R}_A\mathbf{R}_X \in SO(3)$ is arbitrary. Now, if $(\mathbf{X}, \mathbf{Y})$ is a solution to Equation (3.2.2), then there exists some $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2 \in SO(3)$ such that $\mathbf{R}_X = \mathbf{R}_{A_1}^T\boldsymbol{\Theta}_1 = \mathbf{R}_{A_2}^T\boldsymbol{\Theta}_2$ and $\mathbf{R}_Y = \boldsymbol{\Theta}_1\mathbf{R}_{B_1}^T = \boldsymbol{\Theta}_2\mathbf{R}_{B_2}^T$. Eliminating $\boldsymbol{\Theta}_1$,

$$\boldsymbol{\Theta}_2\mathbf{R}_{B_2}^T\mathbf{R}_{B_1}\boldsymbol{\Theta}_2^T = \mathbf{R}_{A_1}\mathbf{R}_{A_2}^T.$$

Taking the logarithm of both sides, $\boldsymbol{\Theta}_2[\boldsymbol{\beta}]\boldsymbol{\Theta}_2^T = [\boldsymbol{\alpha}]$, or equivalently, $\boldsymbol{\Theta}_2\boldsymbol{\beta} = \boldsymbol{\alpha}$, where $[\boldsymbol{\alpha}] = \log(\mathbf{R}_{A_1}\mathbf{R}_{A_2}^T)$ and $\boldsymbol{\beta} = \log(\mathbf{R}_{B_2}^T\mathbf{R}_{B_1})$. As detailed in [24], a solution exists only if $\|\boldsymbol{\alpha}\| = \|\boldsymbol{\beta}\|$, and is given by the one-parameter family

$$\boldsymbol{\Theta}_2 = \exp([\boldsymbol{\alpha}]t)\boldsymbol{\Theta}_p = \boldsymbol{\Theta}_p\exp([\boldsymbol{\beta}]t),$$

where $t \in [0, 2\pi]$ and $\boldsymbol{\Theta}_p$ is any particular solution to $\boldsymbol{\Theta}\boldsymbol{\beta} = \boldsymbol{\alpha}$. Substituting into $\mathbf{R}_X = \mathbf{R}_{A_2}^T\boldsymbol{\Theta}_2$ and $\mathbf{R}_Y = \boldsymbol{\Theta}_2\mathbf{R}_{B_2}^T$ leads to the main result. □

A.1.2 Proof of Proposition 3.2

Proof. We first prove the forward direction. Setting $\mathbf{\Theta}_i = \mathbf{R}_{A_i}\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_{B_i}$, $i = 1, 2, 3$, we have four equations associated with $\mathbf{R}_X = \mathbf{R}_{A_i}^T\mathbf{\Theta}_i$ and $\mathbf{R}_Y = \mathbf{\Theta}_i\mathbf{R}_{B_i}^T$:

$$\begin{aligned} \mathbf{R}_{A_1}^T\mathbf{\Theta}_1 &= \mathbf{R}_{A_2}^T\mathbf{\Theta}_2 &= \mathbf{R}_{A_3}^T\mathbf{\Theta}_3 \\ \mathbf{\Theta}_1\mathbf{R}_{B_1}^T &= \mathbf{\Theta}_2\mathbf{R}_{B_2}^T &= \mathbf{\Theta}_3\mathbf{R}_{B_3}^T. \end{aligned}$$

After some manipulation, $\mathbf{\Theta}_1\mathbf{R}_{B_1}^T\mathbf{R}_{B_j}\mathbf{\Theta}_1^T = \mathbf{R}_{A_j}\mathbf{R}_{A_1}^T, \quad j = 2, 3$. Taking the logarithm of both sides,

$$\mathbf{\Theta}_1\boldsymbol{\beta}_{j1} = \boldsymbol{\alpha}_{j1}, \quad j = 2, 3. \tag{A.1.1}$$

Adding the following independent equation $\mathbf{\Theta}_1(\boldsymbol{\beta}_{21} \times \boldsymbol{\beta}_{31}) = \boldsymbol{\alpha}_{21} \times \boldsymbol{\alpha}_{31}$ leads to $\mathbf{\Theta}_1\mathbf{\Psi} = \mathbf{\Phi}$. Under our assumptions about $\mathbf{\Phi}$ and $\mathbf{\Psi}$ the solution $\mathbf{\Theta}_1$ is given by $\mathbf{\Theta}_1 = \mathbf{\Phi}\mathbf{\Psi}^{-1}$. It is straightforwardly verified that $\det\mathbf{\Theta}_1 = 1$ and $\mathbf{\Theta}_1^T\mathbf{\Theta}_1 = I$, and therefore a rotation matrix as required.

We now prove the reverse direction. Suppose Φ is singular, in which case $\det\mathbf{\Phi} = \|\boldsymbol{\alpha}_{21} \times \boldsymbol{\alpha}_{31}\|^2 = 0$, or equivalently, $\boldsymbol{\alpha}_{21} = c\boldsymbol{\alpha}_{31}$ for some constant $c \in \mathbb{R}$. For a particular solution $\mathbf{\Theta}_1$ to Equation (A.1.1), $\exp([\boldsymbol{\alpha}_{31}])\mathbf{\Theta}_1$ is also a solution and it follows that the solution is not unique. Likewise, in a similar way it can be shown that $\mathbf{\Psi}$ must be nonsingular in order for the solution to be unique. Finally, Suppose $\mathbf{\Phi}^T\mathbf{\Phi} \neq \mathbf{\Psi}^T\mathbf{\Psi}$. For nonsingular $\mathbf{\Psi}$, no solution exists since $\mathbf{\Theta}_1 = \mathbf{\Phi}\mathbf{\Psi}^{-1}$ does not satisfy $\mathbf{\Theta}_1^T\mathbf{\Theta}_1 = \mathbf{I}$. The case of singular $\mathbf{\Psi}$ has already been proven to have multiple solutions. □

A.1.3 Proof of Proposition 3.3

Proof. It is already shown that there exists a one-parameter family of solutions of rotations $(\mathbf{R}_X, \mathbf{R}_Y)$ if $\|\boldsymbol{\alpha}\| = \|\boldsymbol{\beta}\|$ in Proposition 3.1. For any rotation pair solution

$(\mathbf{R}_X, \mathbf{R}_Y)$, $(\mathbf{p}_X, \mathbf{p}_Y)$ satisfies the following equations:

$$\begin{bmatrix} \mathbf{R}_{A_1} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_X \\ \mathbf{p}_Y \end{bmatrix} = \mathbf{R}_Y \mathbf{b}_1 - \mathbf{a}_1$$

$$\begin{bmatrix} \mathbf{R}_{A_2} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_X \\ \mathbf{p}_Y \end{bmatrix} = \mathbf{R}_Y \mathbf{b}_2 - \mathbf{a}_2.$$

Since there are six unknowns with 6 linear equations, there exists a unique solution for $(\mathbf{p}_X, \mathbf{p}_Y)$ if and only if $rank(\mathbb{A}) = 6$. If $rank(\mathbb{A})$ is less than six, we have additionally $6 - rank(\mathbb{A})$ parameters more for the solution of the translation $(\mathbf{p}_X, \mathbf{p}_Y)$. There already exists a one-parameter family of solutions for the rotation $(\mathbf{R}_X, \mathbf{R}_Y)$, so that (3.2.6) has a $(7 - rank(\mathbb{A}))$-parameter family of solutions $(\mathbf{X}, \mathbf{Y})$. □

A.1.4 Proof of Proposition 3.4

Proof. Proposition 3.2 shows that there exists a unique solution $(\mathbf{R}_X, \mathbf{R}_Y)$ if and only if both $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are nonsingular and $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{\Psi}^T\mathbf{\Psi}$, with the solution $\mathbf{R}_Y$ given by $\mathbf{R}_Y = \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{B}_1$. For the unique rotation solution $(\mathbf{R}_X, \mathbf{R}_Y)$, $(\mathbf{p}_X, \mathbf{p}_Y)$ must satisfy the following equations:

$$\begin{bmatrix} \mathbf{R}_{A_i} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_X \\ \mathbf{p}_Y \end{bmatrix} = \mathbf{\Phi}\mathbf{\Psi}^{-1}\mathbf{R}_{B_i}\mathbf{p}_{B_i} - \mathbf{p}_{A_i}, \ i = 1, 2, 3$$

or equivalently $\mathbb{A} \begin{bmatrix} \mathbf{p}_X \\ \mathbf{p}_Y \end{bmatrix} = \boldsymbol{\eta}$. This is a typical over-constrained linear equation with six unknowns. As is well-known, this linear equation has a unique solution if and only if $\mathbb{A}$ has full column rank and $\boldsymbol{\eta}$ is in the range space of $\mathbb{A}$. □

A.2 Derivation of Reduced Objective Function (3.3.9)

Given $\mathbf{X} = \begin{bmatrix} \mathbf{R}_X & \mathbf{p}_X \\ \mathbf{0} & 1 \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} \mathbf{R}_Y & \mathbf{p}_Y \\ \mathbf{0} & 1 \end{bmatrix}$, let us define

$$\boldsymbol{\eta_1} := \begin{bmatrix} \mathbf{r}_X \\ \mathbf{r}_Y \end{bmatrix}, \boldsymbol{\eta_2} := \begin{bmatrix} \mathbf{p}_X \\ \mathbf{p}_Y \end{bmatrix}, \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta_1} \\ \boldsymbol{\eta_2} \end{bmatrix}$$

where $\mathbf{r}_X = vec(\mathbf{R}_X)$, $\mathbf{r}_Y = vec(\mathbf{R}_Y)$. Then (3.3.7) can be expressed as a quadratic function of η:

$$L(\boldsymbol{\eta}) = \frac{1}{2}\boldsymbol{\eta}^T \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{12}^T & \mathbf{H}_{22} \end{bmatrix} \boldsymbol{\eta} + \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}^T \boldsymbol{\eta} + r$$

where $\mathbf{H}_{11} \in \mathbb{R}^{18\times 18}$, $\mathbf{H}_{12} \in \mathbb{R}^{18\times 6}$, $\mathbf{H}_{22} \in \mathbb{R}^{6\times 6}$, $\mathbf{f}_1 \in \mathbb{R}^{18}$, $\mathbf{f}_2 \in \mathbb{R}^6$ and $r \in \mathbb{R}$ are obtained from $(\mathbf{A}_i, \mathbf{B}_i)$ by a straightforward calculation. Since $J(\mathbf{X}, \mathbf{Y})$ is a least squares criterion, this is a convex quadratic minimization with respect to $\boldsymbol{\eta}_2$ for a given $\boldsymbol{\eta}_1$. The minimizer $\boldsymbol{\eta}_2^*$ and the minimum are obtained as follows:

$$\begin{aligned} \boldsymbol{\eta}_2^* &= -\mathbf{H}_{22}^{-1}(\mathbf{H}_{12}^T\boldsymbol{\eta}_1 + \mathbf{f}_2) \\ \min_{\boldsymbol{\eta}_2\in\mathbb{R}^6} L(\boldsymbol{\eta}) &= \frac{1}{2}\boldsymbol{\eta}_1^T\tilde{\mathbf{H}}\boldsymbol{\eta}_1 + \tilde{\mathbf{f}}^T\boldsymbol{\eta}_1 + c \end{aligned} \tag{A.2.2}$$

where $\tilde{\mathbf{H}} = \mathbf{H}_{11} - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{H}_{12}^T$, $\tilde{\mathbf{f}} = \mathbf{f}_1 - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{f}_2$, $c = -\frac{1}{2}\mathbf{f}_2^T\mathbf{H}_{22}^{-1}\mathbf{f}_2 + r$. Here $\tilde{\mathbf{H}}$ is a 18×18 symmetric matrix; eigenvalue decomposition of $\tilde{\mathbf{H}}$ results in an orthogonal eigenvector matrix $\mathbf{X}$ and real-valued eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{18}$:

$$\begin{aligned} \tilde{\mathbf{H}} &= \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^T \\ \boldsymbol{\Lambda} &= diag(\lambda_1, \lambda_2, \ldots, \lambda_{18}) \end{aligned}$$

Now we define 3×3 matrices $(\mathbf{P}_i, \mathbf{Q}_i),\ i = 0, 1, 2, \ldots, 18$ which satisfy

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} vec(\mathbf{P}_1^T) & \ldots & vec(\mathbf{P}_{18}^T) \\ vec(\mathbf{Q}_1^T) & \ldots & vec(\mathbf{Q}_{18}^T) \end{bmatrix} \\ \tilde{\mathbf{f}} &= \begin{bmatrix} vec(\mathbf{P}_0^T) \\ vec(\mathbf{Q}_0^T) \end{bmatrix}. \end{aligned}$$

Since $tr(\mathbf{AB}) = vec(\mathbf{A}^T)vec(\mathbf{B})$ for any matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times 3}$, (A.2.2) is equivalent to

$$J(\mathbf{R}_X, \mathbf{R}_Y) = \frac{1}{2}\sum_{i=1}^{18} \lambda_i \left(tr(\mathbf{P}_i\mathbf{R}_X) + tr(\mathbf{Q}_i\mathbf{R}_Y)\right)^2 + tr(\mathbf{P}_0 R_X) + tr(\mathbf{Q}_0\mathbf{R}_Y) + c$$

where $J(\mathbf{R}_X, \mathbf{R}_Y) = \min_{\boldsymbol{\eta}_2 \in \mathbb{R}^6} L(\boldsymbol{\eta})$.

A.3 Derivations of Gradient and Hessian

$J(\mathbf{R}_X, \mathbf{R}_Y)$ can be expanded up to second order as follows:

$$\begin{aligned} J(\mathbf{R}_X, \mathbf{R}_Y) \approx J(\mathbf{R}_{X_k}, \mathbf{R}_{Y_k}) - \frac{1}{2}\sum_{i=1}^{18} \lambda_i ((tr([\boldsymbol{\alpha}_i][\boldsymbol{\omega}_{\mathbf{R}_X}])^2 + tr([\boldsymbol{\beta}_i][\boldsymbol{\omega}_{\mathbf{R}_Y}])^2 \\ + 2\gamma_i tr([\boldsymbol{\alpha}_i]_k[\boldsymbol{\omega}_{\mathbf{R}_X}]) + \gamma_i tr(\mathbf{M}_i[\boldsymbol{\omega}_{\mathbf{R}_X}]^2) + 2\gamma_i tr([\boldsymbol{\beta}_i][\boldsymbol{\omega}_{\mathbf{R}_Y}]) \\ + \gamma_i tr(\mathbf{N}_i[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2) + 2tr([\boldsymbol{\alpha}_i][\boldsymbol{\omega}_{\mathbf{R}_X}])tr([\boldsymbol{\beta}_i][\boldsymbol{\omega}_{\mathbf{R}_Y}])) - tr([\boldsymbol{\alpha}_0][\boldsymbol{\omega}_{\mathbf{R}_X}]) \\ - tr([\boldsymbol{\beta}_0][\boldsymbol{\omega}_{\mathbf{R}_Y}]) - \frac{1}{2}tr(\mathbf{M}_0[\boldsymbol{\omega}_{\mathbf{R}_X}]^2) - \frac{1}{2}tr(\mathbf{N}_0[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2) \end{aligned}$$

where $[\boldsymbol{\alpha}_i] = \frac{1}{2}(\mathbf{R}_{X_k}^T\mathbf{P}_i^T - \mathbf{P}_i\mathbf{R}_{X_k})$, $[\boldsymbol{\beta}_i] = \frac{1}{2}(\mathbf{R}_{Y_k}^T\mathbf{Q}_i^T - \mathbf{Q}_i\mathbf{R}_{Y_k})$, $\mathbf{M}_i = \frac{1}{2}(\mathbf{R}_{X_k}^T\mathbf{P}_i^T + \mathbf{P}_i\mathbf{R}_{X_k})$, $\mathbf{N}_i = \frac{1}{2}(\mathbf{R}_{Y_k}^T\mathbf{Q}_i^T + \mathbf{Q}_i\mathbf{R}_{Y_k})$, $\gamma_i = tr(\mathbf{M}_i + \mathbf{N}_i)$.

Differentiating the above expansion with respect to $\boldsymbol{\omega}_{\mathbf{R}_X}$ and $\boldsymbol{\omega}_{\mathbf{R}_Y}$, the constant and first-order terms then correspond to the gradient and Hessian, respectively. In this regard, the following proposition, which follows from a straightforward calculation, is useful.

Proposition A.1. *Given* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times 3}$ *and* $[\boldsymbol{\omega}] \in so(3)$,

$$\left[\frac{\partial}{\partial \boldsymbol{\omega}} tr(\mathbf{A}[\boldsymbol{\omega}])\right] = \mathbf{A}^T - \mathbf{A}$$
$$\left[\frac{\partial}{\partial \boldsymbol{\omega}} tr(\mathbf{B}[\boldsymbol{\omega}]^2)\right] = -(\mathbf{B} + \mathbf{B}^T)[\boldsymbol{\omega}] - [\boldsymbol{\omega}](\mathbf{B} + \mathbf{B}^T).$$

Using the above proposition the gradient and Hessian are then given by

$$\nabla J = \begin{bmatrix} \mathbf{s}_X \\ \mathbf{s}_Y \end{bmatrix} \tag{A.3.3}$$

$$\nabla^2 J = -\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^T & \mathcal{C} \end{bmatrix} \tag{A.3.4}$$

where $\mathbf{s}_X = 2\sum_{i=1}^{18} \lambda_i \gamma_i \boldsymbol{\alpha}_i + 2\boldsymbol{\alpha}_0$, $\mathbf{s}_Y = 2\sum_{i=1}^{18} \lambda_i \gamma_i \boldsymbol{\beta}_i + 2\boldsymbol{\beta}_0$, $\mathcal{A} = \sum_{i=1}^{18} \lambda_i (4\boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T + \gamma_i(\mathbf{M}_i - tr(\mathbf{M}_i)\mathbf{I}) + \mathbf{M}_0 - tr(\mathbf{M}_0)\mathbf{I}$, $\mathcal{B} = 4\sum_{i=1}^{18} \lambda_i \boldsymbol{\alpha}_i \boldsymbol{\beta}_i^T$ and $\mathcal{C} = \sum_{i=1}^{18} \lambda_i (4\boldsymbol{\beta}_i \boldsymbol{\beta}_i^T + \gamma_i(\mathbf{N}_i - tr(\mathbf{N}_i)\mathbf{I})) + \mathbf{N}_0 - tr(\mathbf{N}_0)\mathbf{I}$.

A.4 Derivation of Strictly Descending Stepsize Estimate

In Section 3.3 it is stated that once the direction $(\boldsymbol{\omega}_{\mathbf{R}_X}, \boldsymbol{\omega}_{\mathbf{R}_Y})$ is determined, the strictly descending stepsize is given by

$$t^* = -\frac{\phi'(0)}{c}$$

where $\phi'(t) = J(\mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t}, \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t})$ and c is an upper bound of $|\phi''(t)|$. $\phi'(t)$ and $\phi''(t)$ are given as follows:

$$\phi'(t) = \sum_{i=1}^{18} \lambda_i tr(\mathbf{P}_i \mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_i \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}) tr(\mathbf{P}_i \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}] e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_i \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}] e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t})$$
$$+ tr(\mathbf{P}_0 \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}] e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_0 \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}] e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t})$$

$$\phi''(t) = \sum_{i=1}^{18} \lambda_i \left(\left(tr(\mathbf{P}_i \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}] e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_i \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}] e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}) \right)^2 \right.$$
$$\left. + \sum_{i=1}^{18} tr(\mathbf{P}_i \mathbf{R}_{X_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_i \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}) \; tr(\mathbf{P}_i \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}]^2 e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_i \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}]^2 e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}) \right)$$
$$+ tr(\mathbf{P}_0 \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}]^2 e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t} + \mathbf{Q}_0 \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}]^2 e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}). \tag{A.4.5}$$

We remark that the $\mathbf{P}_i, \mathbf{Q}_i$ are constructed from the eigenvectors of $\tilde{\mathbf{H}}$ in (A.2.2) and it follows that, using the maximum eigenvalue of $\tilde{\mathbf{H}}$ (in terms of absolute value), an upper bound for the first sum in (A.4.5) is

$$|\lambda|_{max} \left(\|\mathbf{X}_k [\boldsymbol{\omega}_{\mathbf{R}_X}] e^{[\boldsymbol{\omega}_{\mathbf{R}_X}]t}\|^2 + \|\mathbf{Q}_i \mathbf{R}_{Y_k} e^{[\boldsymbol{\omega}_{\mathbf{R}_Y}]t}\|^2 \right).$$

An upper bound for the second sum in (A.2.2) can be found similarly. Since a linear transformation by a rotation matrix $\mathbf{R} \in SO(3)$ is an isometry (i.e., $\|\mathbf{RU}\| = \|\mathbf{UR}\| = \|\mathbf{U}\|$ for any $\mathbf{U} \in \mathbb{R}^{3\times 3}$), it follows that the strictly descending stepsize is

$$t^* = -\frac{\phi'(0)}{c}$$

where $\phi'(0) = \sum_{i=1}^{18} \lambda_i tr(\mathbf{P}_i \mathbf{R}_{X_k} + \mathbf{Q}_i \mathbf{R}_{Y_k}) tr(\mathbf{P}_i \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}] + \mathbf{Q}_i \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}]) + tr(\mathbf{P}_0 \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}] + \mathbf{Q}_0 \mathbf{R}_{Y_k} [\boldsymbol{\omega}_{\mathbf{R}_Y}])$ and $c = |\lambda|_{max} \left(\|[\boldsymbol{\omega}_{\mathbf{R}_X}]\|^2 + \|[\boldsymbol{\omega}_{\mathbf{R}_Y}]\|^2 \right) + \sqrt{6} |\lambda|_{max} \sqrt{\|[\boldsymbol{\omega}_{\mathbf{R}_X}]^2\|^2 + \|[\boldsymbol{\omega}_{\mathbf{R}_Y}]^2\|^2} + \sqrt{3} \left(\|\mathbf{P}_0 \mathbf{R}_{X_k} [\boldsymbol{\omega}_{\mathbf{R}_X}]^2\| + \|\mathbf{Q}_0 \mathbf{Y}_k [\boldsymbol{\omega}_{\mathbf{R}_Y}]^2\| \right)$.

A.5 Unscented Kalman Filtering on Vector Space

Let us briefly review the standard UKF algorithm on vector space [57, 58]. Consider a state $\mathbf{x}_k \in \mathbb{R}^{N_x}$ on the vector space and the nonlinear dynamical system with discrete-time model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{n}_k) \tag{A.5.6}$$

$$\mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}, \mathbf{w}_{k+1}), \tag{A.5.7}$$

where f and h respectively represent the state transition function and the measurement function, of which forms are nonlinear. In Equation (A.5.7), $\mathbf{y}_{k+1} \in \mathbb{R}^{N_y}$ denotes a measurement vector at time step $k+1$. Noise vectors $\mathbf{n}_k, \mathbf{w}_{k+1}$ are zero-mean Gaussian: $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_k)$ and $\mathbf{w}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_{k+1})$. Here $\mathbf{N}_k$ and $\mathbf{W}_{k+1}$ denote the covariances of $\mathbf{n}_k$ and $\mathbf{w}_{k+1}$, respectively.

A.5.1 Time Update

- From *a priori* state estimate $\hat{\mathbf{x}}_{k|k}$ and its covariance $\mathbf{P}_{k|k}$, we can extract $2N_x+1$ symmetrically distributed sample points $\mathcal{X}_k^{(i)} \in \mathbb{R}^{N_x}$, $(i = 0, \ldots, 2N_x)$ called sigma points as follows:

$$\mathcal{X}_k^{(0)} = \hat{\mathbf{x}}_{k|k} \tag{A.5.8}$$

$$\mathcal{X}_k^{(j)} = \hat{\mathbf{x}}_{k|k} + \gamma \mathbf{s}_j, \quad (j = 1, \ldots, N_x) \tag{A.5.9}$$

$$\mathcal{X}_k^{(j+N_x)} = \hat{\mathbf{x}}_{k|k} - \gamma \mathbf{s}_j, \quad (j = 1, \ldots, N_x) \tag{A.5.10}$$

  where $\mathbf{s}_i \in \mathbb{R}^{N_x}$ is the i^{th} column vector of the lower triangular matrix $\mathbf{S} \in \mathbb{R}^{N_x \times N_x}$. Here, $\mathbf{S}$ can be obtained from Cholesky decomposition of $\mathbf{P}_{k|k} = \mathbf{S}\mathbf{S}^T$. Design parameter γ can be assigned as $\gamma = \sqrt{N_x + \lambda}$, where $\lambda = N_x(\alpha^2 - 1)$ and $0 < \alpha < 1$. For notational simplicity, Equations (A.5.8), (A.5.9), and (A.5.10) can be collectively rewritten as

$$\mathcal{X}_k \doteq (\hat{\mathbf{x}}_{k|k}, \quad \hat{\mathbf{x}}_{k|k} + \gamma \mathbf{s}, \quad \hat{\mathbf{x}}_{k|k} - \gamma \mathbf{s}).$$

- By using Equation (A.5.6), we can define

$$\Upsilon_{k+1}^{(i)} := f(\mathcal{X}_k^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 2N_x).$$

The propagated mean and covariance are given by

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2N_x} w_m^{(i)} \Upsilon_{k+1}^{(i)} \tag{A.5.11}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{2N_x} w_c^{(i)} \mathbf{q}_i \mathbf{q}_i^T + \mathbf{N}_k, \tag{A.5.12}$$

where $\mathbf{q}_i := \Upsilon_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1|k} \in \mathbb{R}^{N_x}$ and weights are

$$w_m^{(0)} = \frac{\lambda}{\lambda + N_x}, \quad w_c^{(0)} = \frac{\lambda}{\lambda + N_x} + (1 - \alpha^2 + \beta) \tag{A.5.13}$$

$$w_m^{(i)} = w_c^{(i)} = \frac{1}{2(\lambda + N_x)}, \quad (i = 1, \ldots, 2N_x). \tag{A.5.14}$$

In Equation (A.5.13), $\beta = 2$ for a Gaussian prior.

- A set of sigma points $\mathcal{X}_{k+1}^{(i)}$, $(i = 0, \ldots, 2N_x)$ is redrawn from $\Upsilon_{k+1}^{(i)}$ by using Equations (A.5.11) and (A.5.12):

$$\mathcal{X}_{k+1} \doteq (\hat{\mathbf{x}}_{k+1|k}, \quad \hat{\mathbf{x}}_{k+1|k} + \gamma \mathbf{u}, \quad \hat{\mathbf{x}}_{k+1|k} - \gamma \mathbf{u}),$$

where $\mathbf{u} \in \mathbb{R}^{N_x}$ denotes the corresponding column vector of the lower triangular matrix $\mathbf{U}$. Here, $\mathbf{P}_{k+1|k} = \mathbf{U}\mathbf{U}^T$.

A.5.2 Measurement Update

- From Equation (A.5.7), we can define a set of measurement sigma points

$$\mathcal{Y}_{k+1}^{(i)} := h(\mathcal{X}_{k+1}^{(i)}, \mathbf{0}), \quad (i = 0, \ldots, 2N_x).$$

The mean and covariance of $\mathcal{Y}_{k+1}^{(i)}, i = 0, \ldots, 2N_x$ are

$$\hat{\mathbf{y}}_{k+1} = \sum_{i=0}^{2N_x} w_m^{(i)} \mathcal{Y}_{k+1}^{(i)} \tag{A.5.15}$$

$$\mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{2N_x} w_c^{(i)} \mathbf{z}_i \mathbf{z}_i^T + \mathbf{W}_{k+1}, \tag{A.5.16}$$

where $\mathbf{z}_i := \mathcal{Y}_{k+1}^{(i)} - \hat{\mathbf{y}}_{k+1} \in \mathbb{R}^{N_y}$. The covariance $\mathbf{P_{xy}}$ can be obtained by

$$\mathbf{P_{xy}} = \sum_{i=0}^{2N_x} w_c^{(i)} \mathbf{p}_i \mathbf{z}_i^T,$$

where $\mathbf{p}_i := \mathcal{X}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1|k} \in \mathbb{R}^{N_x}$.

- The Kalman gain is given by $\mathbf{K} = \mathbf{P_{xy}}\mathbf{P_{yy}^{-1}}$. As a final stage, the state and covariance are updated as

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\boldsymbol{\delta} \tag{A.5.17}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{P_{yy}}\mathbf{K}^T, \tag{A.5.18}$$

where $\boldsymbol{\delta} := \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}$ represents the innovation vector.

A.6 UKF on Matrix Lie Group with Vector Measurements

Consider a state $\mathbf{X}_k$ in an element of a matrix Lie group $\mathcal{G}$ and dynamical system in the discrete-time

$$\mathbf{X}_{k+1} = \mathcal{F}(\mathbf{X}_k, \mathbf{n}_k) \in \mathcal{G} \tag{A.6.19}$$

$$\mathbf{y}_{k+1} = h(\mathbf{X}_{k+1}, \mathbf{w}_{k+1}) \in \mathbb{R}^{N_y}, \tag{A.6.20}$$

where $\mathbf{y}_{k+1}$ is the measurement on vector space. In time update stage, the algorithm is exactly the same as described in Section 4.2.0.1. In measurement update stage, the mean and covariance of the estimated measurement are computed by both Equations (A.5.15) and (A.5.16) given in Section A.5.2. *A posteriori* state and covariance are updated as explained in Section 4.2.0.2.

A.7 Motion and Magnetic Disturbances

In dynamic motions with large accelerations, accelerometers produce the vector sum of the negative gravity and additional accelerations expressed in the body frame $\{\mathcal{B}\}$ fixed to the IMU. Harada [59] calls these additional acceleration terms as a motion disturbance. In magnetically disturbed environments, the measurements of magnetometers are deviated from the local geomagnetic field expressed in $\{\mathcal{B}\}$.

To detect these disturbances, many reliability functions have been proposed [59, 60, 61]. However, for simple and effective detection of disturbances, checking only the norms of calibrated outputs coming from accelerometers or magnetometers is enough [62, 55].

Let us briefly review the measurement reliability function $\tilde{\Psi}(\cdot)$ in [62]. Let $\tilde{\Psi}(\cdot)$ be a binary indicator function; $\tilde{\Psi} : \check{\mathbf{v}}_i \mapsto \{0, 1\}$ where $\check{\mathbf{v}}_i, (i = 1 \,\text{or}\, 2)$ is the calibrated output of accelerometers or magnetometers, which is unnormalized. If $|\|\check{\mathbf{v}}_i\| - 1| < \gamma_i$, then $\tilde{\Psi}(\check{\mathbf{v}}_i) = 0$ (reliable measurement). Otherwise, $\tilde{\Psi}(\check{\mathbf{v}}_i) = 1$ (unreliable measurement). Here, $\gamma_i \in \mathbb{R}$ is a certain threshold. When $\tilde{\Psi}(\check{\mathbf{v}}_i) = 1$, we call that disturbance is detected.

When dealing with motion or magnetic disturbances in probabilistic attitude filters, two methods are commonly used:

- Adaptation of noise covariances [63, 64]: If disturbance is detected, noise covariance of Kalman filter is adjusted.

- Measurement reconstruction with a vector selector [55, 19]: If $\tilde{\Psi}(\check{\mathbf{v}}_i) = 1$, then $\check{\mathbf{v}}_i$ is replaced by $\hat{\mathbf{R}}_{k+1|k}^T \mathbf{r}_i$. Here, $\hat{\mathbf{R}}_{k+1|k}$ is given by Equations (4.3.39) and (4.3.41).

In our attitude filter, the measurement reconstruction method with a vector selector is used for fast response to disturbances.

A.8 Extrinsic Mean of Unit Vectors

Proposition A.2. *Given a set of unit vectors,* $\mathcal{S}_v = \{\mathbf{v}_i \in \mathbb{R}^D | \|\mathbf{v}_i\| = 1, i = 1, \ldots, N\}$*, the extrinsic mean of* $\mathcal{S}_v$ *is defined as* $\mathbf{v}^* := \arg\min_{\mathbf{v}} \sum_{i=1}^{N} \|\mathbf{v}_i - \mathbf{v}\|^2$ *subject to* $\|\mathbf{v}\| = 1$*. If* $\mathbf{m} := \sum_{i=1}^{N} \mathbf{v}_i \neq \mathbf{0}$*, then* $\mathbf{v}^* = \mathbf{m}/\|\mathbf{m}\|$*.*

Proof. We construct the *Lagrangian* as $L(\mathbf{v}, \lambda) = \sum_{i=1}^{N} \|\mathbf{v}_i - \mathbf{v}\|^2 + \lambda(\mathbf{v}^T\mathbf{v} - 1)$ where $\lambda > 0$. The first order necessary conditions ($\frac{\partial L(\mathbf{v}^*, \lambda)}{\partial \mathbf{v}^*} = 0$ and $\frac{\partial L(\mathbf{v}^*, \lambda)}{\partial \lambda} = 0$) yield the result. □

A.9 Proof of Proposition 4.1

Before we begin to prove Proposition 4.1, let us first derive a Jacobian for the solution to Wahba's problem.

A.9.1 Jacobian for the Solution to Wahba's Problem

Given $\mathbf{A} \in SO(3)$ near the inverse of a true attitude, let us consider the following optimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^3} \sum_{i=1}^{2} \frac{1}{\sigma_i^2} \|\mathbf{v}_i - \exp([\boldsymbol{\theta}])\, \mathbf{A}\mathbf{r}_i\|^2, \tag{A.9.21}$$

where $\mathbf{v}_i = \mathbf{A}\mathbf{r}_i + \Delta\mathbf{v}_i$. The solution to the problem described in Equation (A.9.21) can be calculated by

$$\exp([\boldsymbol{\theta}^*]) = \mathbf{V}_p \mathbf{U}_p^T,$$

where $\mathbf{V}_p, \mathbf{U}_p$ are earned from the SVD of the attitude profile matrix, $\mathbf{F}_p := \sum_{i=1}^{2} \frac{1}{\sigma_i^2} \mathbf{A}\mathbf{r}_i \mathbf{v}_i^T = \mathbf{U}_p \mathbf{\Sigma}_p \mathbf{V}_p^T$.

If $\Delta\mathbf{v}_i = \mathbf{0}$, then its attitude profile matrix $\mathbf{F}_0 := \sum_{i=1}^{2} \frac{1}{\sigma_i^2} \mathbf{A}\mathbf{r}_i \mathbf{r}_i^T \mathbf{A}^T = \mathbf{U}_0 \mathbf{\Sigma}_0 \mathbf{V}_0^T$ is symmetric, i.e., $\mathbf{V}_0 = \mathbf{U}_0$, and $\mathbf{V}_0 \mathbf{U}_0^T = \mathbf{I}_3$ yields $\boldsymbol{\theta}^* = \mathbf{0}$. If $\Delta\mathbf{v}_i$ is small, then $\boldsymbol{\theta}^*$ will be small. From the first order approximation, we have

$$\exp([\boldsymbol{\theta}^*]) \approx \mathbf{I}_3 + [\boldsymbol{\theta}^*] \tag{A.9.22}$$

$$\mathbf{V}_p \mathbf{U}_p^T = (\mathbf{V}_0 + \Delta\mathbf{V})(\mathbf{U}_0 + \Delta\mathbf{U})^T, \tag{A.9.23}$$

where $\Delta\mathbf{U}, \Delta\mathbf{V}$ respectively denote deviations from orthogonal matrices $\mathbf{U}_0, \mathbf{V}_0$ due to $\Delta\mathbf{v}_i$ and $\Delta\mathbf{U} \neq \Delta\mathbf{V}$ even if $\mathbf{U}_0 = \mathbf{V}_0$ in general. By assuming $\Delta\mathbf{V}\Delta\mathbf{U}^T \approx \mathbf{0}$ and using the property $\mathbf{V}_0 \mathbf{U}_0^T = \mathbf{I}_3$, Equation (A.9.23) can be rewritten as

$$\mathbf{V}_p \mathbf{U}_p^T \approx \mathbf{I}_3 + \Delta\mathbf{V}\mathbf{U}_0^T + \mathbf{V}_0 \Delta\mathbf{U}^T. \tag{A.9.24}$$

Let $T_{\boldsymbol{\theta}^*} \in \mathbb{R}^3$ denote the estimator of $\boldsymbol{\theta}^*$ as a function of $\Delta\mathbf{v}_i$. By equating the approximations in Equations (A.9.22) and (A.9.24), the estimator $T_{\boldsymbol{\theta}^*}$ can be expressed as

$$[T_{\boldsymbol{\theta}^*}] = \Delta\mathbf{V}\mathbf{U}_0^T + \mathbf{V}_0 \Delta\mathbf{U}^T \in so(3). \tag{A.9.25}$$

After some algebra, $T_{\boldsymbol{\theta}^*}$ is given as the form of a linear function of $\Delta\mathbf{v}_i$ and can be expressed as

$$T_{\boldsymbol{\theta}^*} = \sum_{i=1}^{2} \mathbf{J}_i \Delta\mathbf{v}_i,$$

where $\mathbf{J}_i \in \mathbb{R}^{3\times 3}$ is the Jacobian of $T_{\boldsymbol{\theta}^*}$ with respect to $\Delta\mathbf{v}_i$ at the nominal attitude $\mathbf{A}$. The Jacobian $\mathbf{J}_i$ is given as follows:

$$\mathbf{J}_i = \frac{1}{\sigma_i^2} \mathbf{U}_0 \tilde{\mathbf{D}} \mathbf{U}_0^T [\mathbf{A}\mathbf{r}_i], \tag{A.9.26}$$

where $\tilde{\mathbf{D}} = \text{diag}(1/(s_2+s_3), 1/(s_3+s_1), 1/(s_1+s_2))$ and $s_j, (j=1,2,3)$ represent the singular values of $\mathbf{F}_0$. Since $\mathbf{F}_0 = \mathbf{U}_0\boldsymbol{\Sigma}_0\mathbf{U}_0^T$ and $tr(\mathbf{F}_0) = tr(\boldsymbol{\Sigma}_0) = \sum_{j=1}^{3} s_j = \sum_{j=1}^{2} \frac{1}{\sigma_j^2}$, we have

$$\mathbf{U}_0\tilde{\mathbf{D}}\mathbf{U}_0^T = (tr(\boldsymbol{\Sigma}_0)\mathbf{I}_3 - \mathbf{U}_0\boldsymbol{\Sigma}_0\mathbf{U}_0^T)^{-1} \tag{A.9.27}$$

$$= (\sum_{i=1}^{2} \frac{1}{\sigma_i^2}(\mathbf{I}_3 - \mathbf{A}\mathbf{r}_i\mathbf{r}_i^T\mathbf{A}^T))^{-1}, \tag{A.9.28}$$

where $tr(\cdot)$ denotes the trace of a matrix. By substituting Equation (A.9.28) into (A.9.26), we can obtain

$$\mathbf{J}_i = \mathbf{M}^{-1}(\frac{1}{\sigma_i^2}[\mathbf{A}\mathbf{r}_i]), \tag{A.9.29}$$

where

$$\mathbf{M} := \sum_{i=1}^{2} \frac{1}{\sigma_i^2}(\mathbf{I}_3 - \mathbf{A}\mathbf{r}_i\mathbf{r}_i^T\mathbf{A}^T). \tag{A.9.30}$$

From Equation (A.9.29), one can notice that the weighting factor $1/\sigma_i^2$ of each measurement is reflected in $\mathbf{J}_i$. Note that $\mathbf{M}$ in Equation (A.9.30) is the Fisher information matrix $F_{\boldsymbol{\theta}^*\boldsymbol{\theta}^*} = E(-\frac{\partial^2 \log(p(\mathbf{v}_1,\mathbf{v}_2;\boldsymbol{\theta}^*))}{\partial \boldsymbol{\theta}^{*2}})$, where the likelihood function is defined as

$$p(\mathbf{v}_1, \mathbf{v}_2; \boldsymbol{\theta}^*) := \eta_1 \exp(-\sum_{i=1}^{2} \|\mathbf{v}_i - \exp([\boldsymbol{\theta}^*])\mathbf{A}\mathbf{r}_i\|^2/2\sigma_i^2).$$

The $\boldsymbol{\theta}^*$ terms will be replaced by $\boldsymbol{\theta}$ in the next section for notational simplicity.

A.9.2 Proof of Proposition 4.1

From the theorem of CRLB, the covariance of an unbiased estimator $T_{\boldsymbol{\theta}}$ is lower bounded by the inverse of Fisher information matrix [65]

$$\text{cov}(T_{\boldsymbol{\theta}}) \geq F_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1}.$$

The necessary and sufficient condition for the equality is given as

$$\frac{\partial \log(p)}{\partial \boldsymbol{\theta}} = F_{\boldsymbol{\theta}\boldsymbol{\theta}}(T_{\boldsymbol{\theta}} - \boldsymbol{\theta}), \tag{A.9.31}$$

where $T_{\boldsymbol{\theta}}$ denotes the unbiased estimator and p is the likelihood function. Therefore it is enough to check if $T_{\boldsymbol{\theta}}$ given by Equation (A.9.25) is truly the unbiased estimator and if this equality is true for the proof of the satisfaction of the CRLB.

The mean of the estimator $T_{\boldsymbol{\theta}}$ can be computed by

$$E(T_{\boldsymbol{\theta}}) = E(\sum_{i=1}^{2} \mathbf{J}_i \Delta \mathbf{v}_i) = \sum_{i=1}^{2} \mathbf{J}_i E(\Delta \mathbf{v}_i) \tag{A.9.32}$$

$$= \sum_{i=1}^{2} \mathbf{J}_i (\exp([\boldsymbol{\theta}]) - \mathbf{I}_3)\mathbf{A}\mathbf{r}_i \tag{A.9.33}$$

$$\approx -\sum_{i=1}^{2} \mathbf{J}_i [\mathbf{A}\mathbf{r}_i]\boldsymbol{\theta} = -\mathbf{M}^{-1}(\sum_{i=1}^{2} \frac{1}{\sigma_i^2}[\mathbf{A}\mathbf{r}_i]^2)\boldsymbol{\theta} \tag{A.9.34}$$

$$= \mathbf{M}^{-1}\mathbf{M}\boldsymbol{\theta} = \boldsymbol{\theta}, \tag{A.9.35}$$

where we used the fact that $E(\mathbf{v}_i) = \exp([\boldsymbol{\theta}])\mathbf{A}\mathbf{r}_i$ for Equation (A.9.33), and $[\mathbf{a}]^2 = \mathbf{a}\mathbf{a}^T - \|\mathbf{a}\|^2\mathbf{I}_3$ for any $\mathbf{a} \in \mathbb{R}^3$ in Equation (A.9.34). Hence the estimator $T_{\boldsymbol{\theta}}$ is unbiased under the first order approximation.

The log-likelihood and its derivative with respect to estimation parameter $\boldsymbol{\theta}$ is given as follows:

$$-\log(p) = \sum_{i=1}^{2} \frac{1}{2\sigma_i^2} \|\mathbf{v}_i - \exp([\boldsymbol{\theta}])\mathbf{A}\mathbf{r}_i\|^2 \tag{A.9.36}$$

$$\frac{\partial \log(p)}{\partial \boldsymbol{\theta}} \approx \sum_{i=1}^{2} \frac{1}{\sigma_i^2} \left([\mathbf{A}\mathbf{r}_i]^2\boldsymbol{\theta} + [\mathbf{A}\mathbf{r}_i](\mathbf{v}_i - \mathbf{A}\mathbf{r}_i)\right) \tag{A.9.37}$$

$$= \sum_{i=1}^{2} \left(\frac{1}{\sigma_i^2}[\mathbf{A}\mathbf{r}_i]\Delta \mathbf{v}_i\right) - F_{\boldsymbol{\theta}\boldsymbol{\theta}}\boldsymbol{\theta}. \tag{A.9.38}$$

In Equation (A.9.37), the first order approximation $\exp([\boldsymbol{\theta}]) \approx \mathbf{I}_3 + [\boldsymbol{\theta}]$ is used. From Equation (A.9.29),

$$\sum_{i=1}^{2} \left(\frac{1}{\sigma_i^2} [\mathbf{A}\mathbf{r}_i] \Delta \mathbf{v}_i \right) = F_{\boldsymbol{\theta}\boldsymbol{\theta}} \sum_{i=1}^{2} \mathbf{J}_i \Delta \mathbf{v}_i = F_{\boldsymbol{\theta}\boldsymbol{\theta}} T_{\boldsymbol{\theta}}. \tag{A.9.39}$$

When substituting Equation (A.9.39) into (A.9.38), we can see that the equality condition (A.9.31) holds under the first order approximation, and therefore CRLB can be achieved for this estimator. This completes the proof.

Bibliography

[1] Junhyoung Ha, Donghoon Kang, and Frank C. Park. A stochastic global optimization algorithm for the two-frame sensor calibration problem. *IEEE Trans. Ind. Electron.*, 63(4):2434–2446, 2016.

[2] Donghoon Kang, Jinwook Kim, and Sung-Kyu Kim. Affine registration of three-dimensional point sets for improving the accuracy of eye position trackers. *Opt. Eng.*, 56(4):43105, 2017.

[3] Hanqi Zhuang, Zvi S. Roth, and R. Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form $AX = YB$. *IEEE Trans. Robot. Autom.*, 10(4):549–554, 1994.

[4] Fadi Dornaika and Radu Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Trans. Robot. Autom.*, 14(4):617–622, 1998.

[5] Aiguo Li, Lin Wang, and Defeng Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product. *Int. J. Physical Sciences*, 10(5):1530–1536, 2010.

[6] John L. Crassidis, F. Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *J. Guid. Control Dyn.*, 30(1):12–28, 2007.

[7] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. IEEE*, 92(3):401–422, 2004.

[8] Chanki Kim, Rathinasamy Sakthivel, and Wan Kyun Chung. Unscented FastSLAM: A robust and efficient solution to the SLAM problem. *IEEE Trans. Robot.*, 24(4):808–820, 2008.

[9] Antonio Giannitrapant, Nicola Ceccarelli, Fabrizio Scortecci, and Andrea Garulli. Comparison of EKF and UKF for spacecraft localization via angle measurements. *IEEE Tran. Aerosp. Electron. Syst.*, 47(1):75–84, 2011.

[10] Guillaume Bourmaud, Rémi Mégret, Marc Arnaudon, and Audrey Giremus. Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions. *J. Math. Imaging and Vision*, 51(1):209–228, 2015.

[11] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. From intrinsic optimization to iterated extended Kalman filtering on Lie groups. *J. Math. Imaging and Vision*, 55(3):284–303, 2016.

[12] Axel Barrau and Silvere Bonnabel. Intrinsic filtering on Lie groups with applications to attitude estimation. *IEEE Trans. Automat. Contr.*, 60(2):436–449, 2015.

[13] Axel Barrau and Silvere Bonnabel. The invariant extended Kalman filter as a stable observer. *IEEE Trans. Automat. Contr.*, 62(4):1797–1812, 2017.

[14] Soren Hauberg, Francois Lauze, and Kim Steenstrup Pedersen. Unscented Kalman filtering on Riemannian manifolds. *J. Math. Imaging and Vision*, 46(1):103–120, 2013.

[15] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[16] Junghyun Kwon, Hee Seok Lee, Frank C. Park, and Kyoung Mu Lee. A geometric particle filter for template-based visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 36(4):625–643, 2014.

[17] John L. Crassidis and F. Landis Markley. Unscented filtering for spacecraft attitude estimation. *J. Guid. Control Dyn.*, 26(4):536–542, 2003.

[18] Lubin Chang and Baiqing Hu. Modified unscented quaternion estimator based on quaternion averaging. *J. Guid. Control Dyn.*, 37(1):305–308, 2014.

[19] Jung Keun Lee and Edward J. Park. Minimum-order Kalman filter with vector selector for accurate estimation of human body orientation. *IEEE Trans. Robot.*, 25(5):1196–1201, 2009.

[20] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Automat. Contr.*, 53(5):1203–1218, 2008.

[21] A.H.G. Rinnooy Kan and G.T. Timmer. Stochastic global optimization methods part i: Clustering methods. *Mathematical Programming*, 39(1):27–56, 1987.

[22] A.H.G. Rinnooy Kan and G.T. Timmer. Stochastic global optimization methods part i: Multi level methods. *Mathematical Programming*, 39(1):57–78, 1987.

[23] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, 2nd ed.* Springer, Switzerland, 2016.

[24] Frank C. Park and Bryan J. Martin. Robot sensor calibration: Solving $AX = XB$ on the Euclidean group. *IEEE Trans. Robot. Autom.*, 10(5):717–721, 1994.

[25] Frank C. Park. Computational aspects of the product-of-exponentials formula for robot kinematics. *IEEE Trans. Automat. Contr.*, 39(3):643–647, 1994.

[26] Timothy D. Barfoot and Paul T. Furgale. Associating uncertainty with three dimensional poses for use in estimation problems. *IEEE Trans. Robot.*, 30(3):679–693, 2014.

[27] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, 1994.

[28] Philippe Dreesen, Kim Batselier, and Bart De Moor. Back to the roots: Polynomial system solving, linear algebra, systems theory. Proc. 16th IFAC Symp. on System Identification, 2012.

[29] Anna Yershova, Steven M. LaValle, and Julie C. Mitchell. Generating uniform incremental grids on $so(3)$ using the hopf fibration. Proc. Workshop on Algorithmic Foundations of Robotics, 2008.

[30] Persi Diaconis and Mehrdad Shahshahani. The subgroup algorithm for generating uniform random variables. *Problems in Engineering and Information Sciences*, 1:15–32, 1987.

[31] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

[32] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *Int. J. Robot. Res.*, 18(3):286–298, 1999.

[33] Jung-Young Son, Wook-Ho Son, Sung-Kyu Kim, Kwang-Hoon Lee, and Bahram Javidi. Three-dimensional imaging for creating real-world-like environments. *Proc. IEEE*, 101(1):190–205, 2013.

[34] Donghyun Kim, Hyoung Lee, Sung-Kyu Kim, and Kwanghoon Sohn. Effect of parallax distribution and crosstalk on visual comfort in parallax barrier autostereoscopic display. *Opt. Eng.*, 54(5):053107, 2015.

[35] Donghyun Kim, Hyoung Lee, Sung-Kyu Kim, and Kwanghoon Sohn. Parallax adjustment for visual comfort enhancement using the effect of parallax distribution and cross talk in parallax-barrier autostereoscopic three-dimensional display. *Opt. Eng.*, 54(12):123104, 2015.

[36] N. A. Dodgson. Autostereoscopic 3d displays. *IEEE Computer*, 38(8):31–36, 2005.

[37] Sung-Kyu Kim, Ki-Hyuk Yoon, Seon Kyu Yoon, and Heongkyu Ju. Parallax barrier engineering for image quality improvement in an autostereoscopic 3D display. *Opt. Express*, 23(10):13230–13244, 2015.

[38] Sung-Kyu Kim, Ki-Hyuk Yoon, Seon Kyu Yoon, and Heongkyu Ju. Defragmented image based autostereoscopic 3D displays with dynamic eye tracking. *Opt.Commun.*, 357(15):185–192, 2015.

[39] Aurora Fassi, Marco Riboldi, Christian Fabio Forlani, and Guido Baroni. Optical eye tracking system for noninvasive and automatic monitoring of eye position and movements in radiotherapy treatments of ocular tumors. *Appl. Opt.*, 51(13):13230–13244, 2012.

[40] Zhiwei Zhu and Qiang Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE Trans. Biomed. Eng.*, 54(12):2246–2260, 2007.

[41] Nicole M. Bakker, Boris A. J. Lenseigne, Sander Schutte, Elsbeth B. M. Geukers, Pieter P. Jonker, Frans C. T. van der Helm, and Huibert J. Simonsz. Accurate gaze direction measurements with free head movement for strabismus angle estimation. *IEEE Trans. Biomed. Eng.*, 60(11):3028–3035, 2013.

[42] Hyeon Chang Lee, Duc Thien Luong, Chul Woo Cho, Eui Chul Lee, and Kang Ryoung Park. Gaze tracking system at a distance for controlling IPTV. *IEEE Trans. Consum. Electro.*, 56(4):2577–2583, 2010.

[43] Yoshinobu Ebisawa and Kiyotaka Fukumoto. Long–range gaze tracking system for large movements. *IEEE Trans. Biomed. Eng.*, 60(12):3432–3440, 2013.

[44] Shiyu Song and Manmohan Chandraker. Robust scale estimation in real–time monocular SFM for autonomous driving. IEEE Conf. CVPR, pages 23–28, June 2014.

[45] P. Jonathon Phillips, W. Todd Scruggs, Alice J. O'Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, and Matthew Sharpe. Fast, reliable head tracking under varying illumination: An approach based on registration of texture–mapped 3D models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(4):322–336, 2000.

[46] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Trans. Intell. Transp. Syst.*, 11(2):300–311, 2010.

[47] Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Trans. Image Process.*, 21(2):802–815, 2012.

[48] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):607–626, 2010.

[49] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, 2010.

[50] Malcom D. Shuster. Maximum likelihood estimation of spacecraft attitude. *J. Astronaut. Sci.*, 37(1):79–88, 1989.

[51] Malcom D. Shuster. The generalized Wahba problem. *J. Astronaut. Sci.*, 54(2):245–259, 2006.

[52] F. Landis Markley. Attitude determination using vector observations and the singular value decomposition. *J. Astronaut. Sci.*, 36(3):245–258, 1988.

[53] Hector Garcia de Marina, Fernando J. Pereda, Jose M. Giron-Sierra, and Felipe Espinosa. UAV attitude estimation using unscented Kalman filter and TRIAD. *IEEE Trans. Ind. Electron.*, 59(11):4465–4474, 2012.

[54] Young Soo Suh. Orientation estimation using a quaternion–based indirect Kalman filter with adaptive estimation of external acceleration. *IEEE Trans. Instrum. Meas.*, 59(12):3296–3305, 2010.

[55] Jung Keun Lee and Edward J. Park. A fast quaternion-based orientation optimizer via virtual rotation for human motion tracking. *IEEE Trans. Biomed. Eng.*, 56(5):1574–1582, 2009.

[56] Myung Hwangbo, Jun-Sik Kim, and Takeo Kanade. Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation. *Int. J. Robot. Res.*, 30(14):1755–1774, 2011.

[57] Eric A. Wan and Rudolph van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proc. Symp. Adaptive Syst. Signal Process. Commun. Contr.*, Lake Louise, Canada, Oct. 2000.

[58] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, Massachusetts, 2005.

[59] Tatsuya Harada, Taketoshi Mori, and Tomomasa Sato. Development of a tiny orientation estimation device to operate under motion and magnetic disturbance. *Int. J. Robot. Res.*, 26(6):547–559, 2007.

[60] Aida Makni, Hassen Fourati, and Alain Y. Kibangou. Energy-aware adaptive attitude estimation under external acceleration for pedestrian navigation. *IEEE/ASME Trans. Mechatron.*, 21(3):1366–1375, 2016.

[61] Riccardo Costanzi, Francesco Fanelli, Niccolo Monni, Alessandro Ridolfi, and Benedetto Allotta. An attitude estimation algorithm for mobile robots under unknown magnetic disturbances. *IEEE/ASME Trans. Mechatron.*, 21(4):1900–1911, 2016.

[62] Henrik Rehbinder and Xiaoming Hu. Drift-free attitude estimation for accelerated rigid bodies. *Automatica*, 40:653–659, 2004.

[63] Angelo M. Sabatini. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Trans. Biomed. Eng.*, 53(7):1346–1356, 2006.

[64] Zongwei Wu, Minli Yao, Hongguang Ma, Weimin Jia, and Fanghao Tian. Low-cost antenna attitude estimation by fusing inertial sensing and two-antenna gps for vehicle-mounted satcom-on-the-move. *IEEE Trans. Veh. Technol.*, 62(3):1084–1096, 2013.

[65] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory, vol.1.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

국문 초록

본 논문에서는 로봇에 다양한 센서를 부착하였을 때 발생하는 캘리브레이션 문제와 상태 추정 문제를 다룬다. 좀 더 구체적으로는 (1) 두 개의 프레임 캘리브레이션 문제와 (2) 관성센서를 이용한 회전행렬과 자이로 바이어스 추정 문제를 다룬다. 상기 문제를 해결하기 위해 기하학적인 알고리즘을 제시하였으며, 인간형 로봇과 무인 비행기 등을 이용한 실제 실험을 통해 제안하는 알고리즘 성능의 우수성을 검증하였다.

두 개의 프레임 캘리브레이션 문제는 움직이는 강체에 고정된 센서 간의 상대적인 좌표계 (프레임) 변환과 움직이지 않는 지면에 고정된 기준 좌표계 간의 상대적인 좌표계 변환을 구하는 것을 말한다. 이 문제를 해결하기 위해 국소 및 전역 최적화를 위한 기하학적인 알고리즘을 제안하였다. 특히, 목적 함수의 gradient, Hessian, 그리고 step size를 해석적인 수식 형태로 제시하여 계산 속도와 정밀도를 향상할 수 있었다.

또한, 관성센서를 이용한 회전행렬과 자이로 바이어스를 실시간 추정하는 기하학적인 알고리즘을 제시하였다. 종래의 벡터 형태의 측정 함수와 달리, Wahba 문제를 통해 계산된 회전 행렬 그 자체를 측정 행렬로 설계하였다. 이를 통해, 측정 노이즈 공분산을 해석적으로 계산할 수 있으며 정밀한 필터링이 가능하게 되는 장점이 있다. 실제 가속도 센서와 지자계 센서를 통해 측정 노이즈 공분산을 계산하는 방법 또한 제시하였다.

주요어: 캘리브레이션, 최적화, 기하학적 추정, 관성 센서

학번: 2012-30167